Quantum algorithms and complexity for certain continuous and related discrete problems

Marek Kwas

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2005

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **Quantum algorithms and complexity for certain continuous and related discrete problems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Columbia University,Graduate School of Arts and Sciences,New York,NY,10027**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**123**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

©2005
Marek Kwas
All Rights Reserved

ABSTRACT

Quantum algorithms and complexity for certain continuous and related discrete problems

Marek Kwas

The thesis contains an analysis of two computational problems. The first problem is discrete quantum Boolean summation. This problem is a building block of quantum algorithms for many continuous problems, such as integration, approximation, differential equations and path integration. The second problem is continuous multivariate Feynman-Kac path integration, which is a special case of path integration.

The quantum Boolean summation problem can be solved by the quantum summation (**QS**) algorithm of Brassard, Høyer, Mosca and Tapp, which approximates the arithmetic mean of a Boolean function. We improve the error bound of Brassard et al. for the worst-probabilistic setting. Our error bound is sharp. We also present new sharp error bounds in the average-probabilistic and worst-average settings. Our average-probabilistic error bounds prove the optimality of the **QS** algorithm for a certain choice of its parameters. The study of the worst-average error shows that the **QS** algorithm is not optimal in this setting; we need to use a certain number of repetitions to regain its optimality.

The multivariate Feynman-Kac path integration problem for smooth multivariate functions suffers from the *provable* curse of dimensionality in the worst-case deterministic setting, i.e., the minimal number of function evaluations needed to compute an approximation depends exponentially on the number of variables. We show that in both the randomized and quantum settings the curse of dimensionality is vanquished,

i.e., the minimal number of function evaluations and/or quantum queries required to compute an approximation depends only polynomially on the reciprocal of the desired accuracy and has a bound independent of the number of variables. The exponents of these polynomials are 2 in the randomized setting and 1 in the quantum setting. These exponents can be lowered at the expense of the dependence on the number of variables. Hence, the quantum setting yields exponential speedup over the worst-case deterministic setting, and quadratic speedup over the randomized setting.

Contents

Acknowledgments vii

1 Introduction **1**

1.1 Background 2

1.1.1 Quantum computation model 4

1.2 Research results 7

1.2.1 Quantum Boolean summation in various settings 7

1.2.1.1 Worst-probabilistic error 9

1.2.1.2 Average-probabilistic error 10

1.2.1.3 Worst-average error 11

1.2.1.4 Quantum Boolean summation simulation 13

1.2.2 Multivariate Feynman-Kac path integration 13

1.2.2.1 Worst-case deterministic setting 15

1.2.2.2 Randomized and quantum settings 16

2 Quantum Boolean summation **19**

2.1 Introduction 19

2.2 Quantum summation algorithm 23

2.3 Performance analysis 32

2.3.1 Worst-probabilistic error 34

2.3.2 Average-probabilistic error 48

2.3.3 Worst-average error 56

2.3.3.1 Local average error 56

2.3.3.2 Quantum summation algorithm with repetitions . . . 65

2.4 Simulation 68

2.4.1 **QS** algorithm simulation 68

2.4.2 Computation of the final state distribution 75

2.5 Conclusions 75

3 Multivariate Feynman-Kac path integration 77

3.1 Introduction 77

3.2 Computational problem 79

3.2.1 Worst-case deterministic setting 80

3.2.2 Randomized setting 80

3.2.3 Quantum setting 81

3.3 The function class $\mathcal{B}_F$ 83

3.4 Series of multivariate integrals 85

3.5 Approximating one term of the series 86

3.5.1 Uniform approximation by Smolyak's algorithm 86

3.5.2 Deterministic algorithm 87

3.5.3 Variance reduction 88

3.5.4 Randomized algorithm 88

3.5.5 Quantum algorithm 89

3.6 Complete algorithms 91

3.6.1 Deterministic algorithm 92

3.6.2 Randomized algorithm 93

3.6.3 Quantum algorithm 94

3.7 Complexity of Feynman-Kac path integration 95

3.7.1 Lower bounds 95

3.7.2 Upper bounds 96
3.8 Examples 97
3.8.1 The Sobolev space of compactly supported functions 97
3.8.2 Periodic functions 98
3.9 Conclusions 99

4 Open problems 101

4.1 Average-average error for quantum Boolean summation 101
4.2 Multivariate Feynman-Kac path integration in the quantum setting with deterministic queries 102
4.3 Feynman-Kac integrals and finite-order weights 103

Bibliography 103

List of Figures

2.1 The estimate of $C(p)$. 45

2.2 The function v on $[0,1]$. 46

2.3 `QSsimul` output for $a_f \simeq \frac{2}{3}$ 73

2.4 `QSsimul` output for $a_f = \frac{7}{8}$ 74

2.5 `QSsimul` output for $a_f = \frac{1}{8}$ 74

Acknowledgments

I wish to express my deep gratitude to my advisor Professor Henryk Woźniakowski for his invaluable help during the time I spent in the PhD program. I am grateful to all other members of the IBC and quantum computation groups at the Columbia University, Professor Joseph F. Traub, Professor Alfred V. Aho, Dr. Anargyros Papageorgiou, Professor Arthur G. Werschulz, and my fellow colleagues Arvid J. Bessen and Krysta Svore for their support and numerous research discussions.

I had the privilege of interacting with the members of the worldwide IBC community during many stimulating meetings. The ideas of Professors Stefan Heinrich, Erich Novak, Leszek Plaskota, Klaus Ritter and Grzegorz W. Wasilkowski were particularly inspiring for my research. I also appreciate discussions with Professors Jakob Creutzig, Bolesław Z. Kacewicz and Shu Tezuka.

This research was supported in part by the National Science Foundation and by the Defense Advanced Research Agency and Air Force Research Laboratory under agreement F30602-01-2-0523.

To my mother Maria Kwas.

Chapter 1

Introduction

This thesis contains an analysis of two computational problems. One of them, Boolean summation, is discrete and the other, multivariate Feynman-Kac path integration, is continuous. As we shall see, they are closely related if we study them in the quantum model of computation. Quantum Boolean summation is a building block of quantum algorithms for many continuous problems, such as integration, approximation, differential equations and path integration. The quantum Boolean summation problem was previously studied in the worst-probabilistic setting. We improve the existing results and extend the analysis to two more settings—average-probabilistic and worst-average. This extension is especially important for the analysis of multivariate Feynman-Kac path integration, in which the concept of randomized queries is used. The thorough knowledge of quantum Boolean summation properties is necessary to establish sharp complexity bounds in the quantum setting with randomized queries.

This chapter is organized as follows. Section 1.1 gives an overview of research in the area of quantum computation for continuous problems. The quantum model of computation for continuous problems with deterministic and randomized queries is presented in Section 1.1.1. The main results of this thesis are outlined in Section 1.2.

1.1 Background

Feynman [12] was the first to suggest computational devices based on the quantum mechanical principles. He conjectured that the intrinsic difficulties of simulating quantum phenomena on classical computers might be overcome by using quantum computational devices. Indeed, the computational intractability of some quantum phenomena caused by exponential requirements of time and memory in the number of simulated components makes classical computers useless for simulating such phenomena.

Deutsch [8] first studied rigorous models of quantum computation. Initially, advantages of quantum computation over classical computation were shown for some discrete problems, e.g., the Deutsch and Deutsch-Jozsa problems of distinguishing between constant and balanced Boolean functions [8, 9] and the Simon problem of checking Boolean function invariance [43].

Shor and Grover made significant contributions that initiated the explosion of research in the area of quantum computation. Shor [42] found a quantum algorithm that solves the problem of factorizing a composite n-bit integer with cost of order $n^2 \log n \log\log n$, which provided an exponential speedup over the best classical algorithm known. Grover [13] discovered a quantum algorithm that solves the problem of searching an unstructured database consisting of N elements with cost of order $\sqrt{N}$, which yields a quadratic speedup over any classical algorithm.

Novak and Heinrich initiated the study of continuous problems in the quantum setting in the framework of information-based complexity. Novak [34] studied the quantum complexity of the multivariate integration problem for Hölder classes and proved exponential speedup over the worst-case deterministic setting and roughly quadratic speedup over the randomized setting. Heinrich [15] extended the quantum computation model to continuous problems and dealt with integration in L_p spaces, showing results similar to [34]. A selection of continuous problems has been considered since then. A partial list includes:

- approximation for Korobov spaces [35],
- approximation for Sobolev spaces [17, 18],
- the eigenvalue problem [1, 22] and the Sturm-Liouville eigenvalue problem [37],
- integration for Sobolev spaces [16, 20, 21],
- ordinary differential equations [23],
- parametric integration [51],
- path integration [48].

The quantum setting gives an exponential speedup over the worst-case deterministic setting, and a quadratic speedup over the randomized setting for integration over Sobolev spaces of multivariate functions. The same is true for path integration. For parametric integration, quantum algorithms turn out to be much faster than the classical ones for some choices of the problem parameters. However, there is no improvement for some other choices of these parameters. A similar situation occurs for approximation of functions. In particular, for Korobov spaces we obtain polynomial speedups over the known algorithms for the classical settings. For Sobolev spaces the complexity in the classical and quantum settings is known. For some choices of the problem parameters we have polynomial speedup; for some other choices there is no speedup. For ordinary differential equations we again have polynomial speedup.

The power of the quantum setting for the eigenvalue problem and the Sturm-Liouville eigenvalue problem depends on what kind of quantum queries are permitted. For bit queries, which are used for the other problems mentioned in the previous paragraph, we have at most polynomial speedups over the classical randomized setting. However, if power queries, see [3, 37, 38], are permitted we have exponential speedup over even the randomized setting for the eigenvalue problem. We recall that power queries are used in the phase estimation algorithm. In this thesis we deal with several variations of bit queries and we do not consider power queries at all.

1.1.1 Quantum computation model

We now give a brief overview of a simplified quantum model of computation for continuous problems with deterministic and randomized queries. Bit queries for discrete problems are also outlined in Section 2.2, see [30]. The model with deterministic queries for continuous problems is thoroughly described in [15]. Randomized queries are studied in [52]. We shall use the framework outlined below in the remaining part of this chapter, as well as in Chapter 3, where we study multivariate Feynman-Kac path integration.

We start with a general computational problem formulation. For a given class F of input functions $f : D \to C$ we want to approximate the solution operator

$$S : F \to G,$$

with G being a normed space whose norm is denoted by $\|\cdot\|_G$. We approximate $S(f)$ by a quantum algorithm as described below.

First, we transform a given input function $f \in F$ by using a classical algorithm P_s with s classical function evaluations and obtain

$$\bar{f} = P_s(f) : D \to C.$$

The goal of this first classical step is to prepare an input for actual quantum computation. This can be used, for instance, to achieve variance reduction of the input function f through approximation, as it is done for multivariate Feynman-Kac path integration in Section 3.5.3. Afterwards, we use the transformed function $\bar{f}$ as the input to a quantum algorithm.

We recall that the main part of a quantum algorithm is a sequence of unitary operators acting on the space $\mathcal{H}_k = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$, which is a tensor product of k copies of the two dimensional complex Hilbert space $\mathbb{C}^2$, i.e.,

$$U_n(\bar{f}) = Q_n\, Q_{\bar{f}} Q_{n-1} \cdots Q_1\, Q_{\bar{f}} Q_0. \tag{1.1}$$

Here, $Q_0, Q_1, \ldots, Q_n$ are unitary operators and $Q_{\bar{f}}$ is a quantum query for $\bar{f} \in P_s(F)$. Quantum queries are used to collect information about an input function $\bar{f}$ and play a role analogous to the use of function values in the worst case and randomized settings. In this thesis we use the most commonly studied *bit* quantum queries in the deterministic and randomized forms.

For a Boolean function $g : \{0, 1, \ldots, 2^{k-1} - 1\} \to \{0, 1\}$, the bit query Q_g is defined as

$$Q_g|j\rangle|y\rangle = |j\rangle|y \oplus g(j)\rangle.$$

Here, $|j\rangle \in \mathcal{H}_{k-1}$ and $|y\rangle \in \mathcal{H}_1$, with $\oplus$ denoting the addition modulo 2.

For a real function $h : [0, 1] \to [0, 1]$ the bit query is constructed by taking the most significant bits of the function h evaluated at some points t_j. More precisely, as in [15], the bit query Q_h is of the form

$$Q_h|j\rangle|y\rangle = |j\rangle|y \oplus \beta(h(\tau(j)))\rangle,$$

where the number of qubits is now $k = m' + m''$ and $|j\rangle \in \mathcal{H}_{m'}$ and $|y\rangle \in \mathcal{H}_{m''}$ with some functions $\beta : [0, 1] \to \{0, 1, \ldots, 2^{m''} - 1\}$ and $\tau : \{0, 1, \ldots, 2^{m'} - 1\} \to [0, 1]$. Hence we compute h at the points $t_j = \tau(j) \in [0, 1]$, and then take $\beta(h(t_j))$ which is the m'' most significant bits of $h(t_j)$. The randomized quantum query $Q_h = Q_{h,\omega}$, defined as in [52], depends on a random element $\omega \in \Omega$, which indicates that we compute the values of h at randomized points $t_{j,\omega}$.

Therefore, if we use randomized queries, the unitary operator (1.1) depends on a random element ω and is of the form

$$U_{n,\omega}(\bar{f}) = Q_n \, Q_{\bar{f},\omega} \, Q_{n-1} \cdots Q_1 Q_{\bar{f},\omega} \, Q_0$$

with n being the number of randomized queries. As usual, we assume that the initial state is $|0\rangle$ and we compute the final state

$$|\psi_{\bar{f}}\rangle = U_n(\bar{f})|0\rangle = Q_n Q_{\bar{f}} Q_{n-1} \cdots Q_1 Q_{\bar{f}} Q_0 |0\rangle$$

for deterministic quantum queries, and

$$|\psi_{\bar{f},\omega}\rangle = U_{n,\omega}(\bar{f})|0\rangle = Q_n Q_{\bar{f},\omega} Q_{n-1} \cdots Q_1 Q_{\bar{f},\omega} Q_0 |0\rangle$$

for randomized ones. Then we perform a measurement of the final state and obtain an outcome $j \in \{0, 1, \ldots, 2^k - 1\}$ with probability

$$p_{\bar{f}}(j) = |\langle \psi_{\bar{f}} | j \rangle|^2 \quad \text{or} \quad p_{\bar{f},\omega}(j) = |\langle \psi_{\bar{f},\omega} | j \rangle|^2.$$

Knowing the outcome j we compute the final result on a classical computer as $\phi(j)$ and $\varphi(j)$ for some mappings $\phi, \varphi : \{0, 1, \ldots, 2^k - 1\} \to G$. Thus, the quantum algorithm A_n yields

$$A_n(\bar{f}, j) = \phi(j) \qquad \text{or} \qquad A_{n,\omega}(\bar{f}, j) = \varphi(j).$$

The most commonly used error criterion for quantum algorithms with deterministic queries is the worst-probabilistic error. This is defined for $\rho \in (\frac{1}{2}, 1]$ by

$$e^{\text{wor-pro}}(A_n, P_s, S, \rho) = \sup_{f \in F} \min_{J : \mu(J,f) \geq \rho} \max_{j \in J} \| S(f) - A_n(P_s(f), j) \|_G, \tag{1.2}$$

with $\mu(J, f) = \sum_{j \in J} p_{\bar{f}}(j)$. We consider here the probabilistic error with respect to the quantum algorithm outcomes. In words, for a fixed $f \in F$, we take the maximal error for the best outcomes with probability at least ρ. Then we maximize the probabilistic errors over all input functions from F. In addition to (1.2), we shall consider a wider selection of error criteria for quantum algorithms with deterministic queries in Chapter 2 for the Boolean summation problem.

For quantum algorithms with randomized queries we use the worst-average-average error criterion

$$e^{\text{wor-avg-avg}}(A_n, P_s, S) = \sup_{f \in F} \left(\mathbb{E} \sum_{j=0}^{2^k - 1} p_{\bar{f},\omega}(j) \| S(f) - A_{n,\omega}(P_s(f), j) \|_G^2 \right)^{1/2}, \tag{1.3}$$

where $\mathbb{E}$ is the expectation over the probability space Ω. For a fixed input function $f \in F$ we measure the L_2 average error over the probability space Ω and with respect to the quantum algorithm outcomes. Then we maximize the average errors over all input functions from F. Observe, that if we neglect the expectation $\mathbb{E}$ in (1.3), then the sum denotes the error of the quantum algorithm with respect to the probability of all the outcomes, i.e., we consider the error when the algorithm *succeeds* and when it *fails* according to (1.2). This error criterion is a reasonable choice for problems where the result of an algorithm cannot be verified. Besides, it is a stronger error criterion than (1.2), and therefore by Chebyshev's inequality the results obtained for (1.3) can be extended to (1.2).

As usual we are also interested in the query complexities for the respective error settings, i.e., the minimal numbers of quantum queries (deterministic or randomized) and the minimal number of the classical function evaluations that are needed to guarantee that the error does not exceed ε, i.e.,

$$\begin{aligned} n^{\text{wor-pro}}(\varepsilon, F) &= \min\{s+n : \ \exists\, P_s \ \exists\, A_n \quad e^{\text{wor-pro}}(A_n, P_s) \le \varepsilon\}, \\ n^{\text{wor-avg-avg}}(\varepsilon, F) &= \min\{s+n : \ \exists\, P_s \ \exists\, A_n \quad e^{\text{wor-avg-avg}}(A_n, P_s) \le \varepsilon\}. \end{aligned}$$

The quantum setting with randomized queries is considered in Chapter 3 for the multivariate Feynman-Kac path integration problem.

1.2 Research results

We now outline the results of this thesis, which are proved in Chapters 2 and 3. The results of Section 1.2.1 are published in [19, 30]. The results of Section 1.2.2 are published in [29] and included in [27, 28].

1.2.1 Quantum Boolean summation in various settings

For all the problems mentioned in Section 1.1, except the eigenvalue problems, the results are based on the quantum Boolean summation (**QS**) algorithm of Brassard, Høyer, Mosca and Tapp [4]. The optimality of this algorithm was proved by Nayak and Wu [31], using the *polynomial method* for quantum query lower bounds obtained by Beals, Buhrman, Cleve, Mosca and de Wolf [2]. Because of the significance of the **QS** algorithm for the quantum complexity of continuous problems, it is important to study this algorithm in greater detail. The performance of a quantum algorithm can be studied in various error settings, depending on how the input and output of the algorithms are treated. We may have worst and average behavior with respect to the input, and probabilistic and average with respect to the output. The **QS** algorithm in the worst-probabilistic error setting was studied in [4]. It was shown that its error is of order $1/M$ with M queries.

To define the Boolean summation problem in the framework of Section 1.1.1 we set $D = \{0, 1, \dots, N-1\}$ and $C = \{0, 1\}$. The class F of input functions is just the class $\mathbb{B}_N$ of all Boolean functions defined on an N element set. The solution operator S computes the mean of a Boolean function

$$S(f) = \frac{1}{N} \sum_{i=0}^{N-1} f(i) \qquad \forall f \in F = \mathbb{B}_N. \tag{1.4}$$

The **QS** algorithm of Brassard, Høyer, Mosca and Tapp [4] computes an approximation to the solution operator S. Information regarding the Boolean function is supplied by quantum queries. Suppose that we use $M-1$ quantum queries. Obviously, the only case of interest is when M is much smaller than N. Brassard et al. [4] proved that the worst-probabilistic error of the **QS** algorithm is at most $\pi/M + \pi^2/M^2$, with probability $8/\pi^2 = 0.81\dots$. Nayak and Wu [31] showed that with probability $\rho \in (\frac{1}{2}, 1)$ the error of any quantum algorithm that uses no more than $M-1$ quantum queries is bounded from below by the quantity proportional to M^{-1}. Therefore, the **QS** algorithm enjoys the smallest possible error modulo a factor multiplying M^{-1}. Since the **QS** algorithm has so many applications, as mentioned also in Section 1.1, it is important to check whether the error estimate of Brassard et al. is sharp and how the error decreases if we lower the probability $\rho = 8/\pi^2$ to $\rho > \frac{1}{2}$. It also seems reasonable to study whether the **QS** algorithm retains its optimality with different error criteria.

We recall that the estimate of Brassard et al. corresponds to the worst-probabilistic setting, which is most frequently used in the study of quantum algorithms. We also study the average performance of the **QS** algorithm with respect to a measure on Boolean functions. This is the average-probabilistic setting. In the worst-average setting, we study the worst performance with respect to Boolean functions and the average performance with respect to all outcomes of a quantum algorithm. We add in passing that the worst-average setting is usually used for the study of the classical Monte Carlo algorithm. This setting is also used to establish a lower bound for the quantum summation and integration algorithms that use randomized queries, see [52].

We study error bounds in the worst-probabilistic and average-probabilistic settings with probabilities $\rho \in (1/2, 8/\pi^2]$. If we want to obtain error bounds with higher probability, it is known that it is enough to run the **QS** algorithm a number of times and take the median as the final result, see e.g., [15].

1.2.1.1 Worst-probabilistic error

The worst-probabilistic error of the **QS** algorithm is defined according to (1.2)

$$e^{\text{wor-pro}}(\mathbf{QS}, \rho) = \sup_{f \in \mathbb{B}_N} \min_{J : \mu(J,f) \geq \rho} \max_{j \in J} |S(f) - \mathbf{QS}(f, j)|$$

with S given by (1.4) and $\rho \in (1/2, 1]$. Note that $\bar{f} = f$, i.e., we do not use a classical computer to prepare the input for the **QS** algorithm.

Our results improve the worst-probabilistic error bound $e^{\text{wor-pro}}(\mathbf{QS}, 8/\pi^2) \leq \pi/M + \pi^2/M^2$ of Brassard et al. Namely, the worst-probabilistic error of the **QS** algorithm $e^{\text{wor-pro}}(\mathbf{QS}, 8/\pi^2)$ is at most

$$\frac{3}{4}\frac{\pi}{M} \qquad \text{with probability} \quad \frac{8}{\pi^2} = 0.81\ldots.$$

Furthermore, the last estimate is sharp when M and N/M are large. Indeed, for $\rho \in (1/2, 8/\pi^2]$, the error of the **QS** algorithm is at most

$$\frac{\left(1 - v^{-1}(\rho)\right)\pi}{M} \qquad \text{with probability}\, \rho,$$

where v^{-1} is the inverse of the function $v(\Delta)=\sin^2(\pi\Delta)/(\pi\Delta)^2$. We have $1-v^{-1}(\rho)\in(\frac{1}{2},\frac{3}{4}]$ and it is well approximated by $\frac{1}{16}\pi^2\rho+\frac{1}{4}$. In particular, for the most frequently considered values of ρ we have

$$\big(1-v^{-1}(\tfrac{1}{2}+)\big)\pi=1.75\dots,\quad \big(1-v^{-1}(\tfrac{3}{4})\big)\pi=2.23\dots,$$
$$\big(1-v^{-1}(8/\pi^2)\big)\pi=\tfrac{3}{4}\pi=2.35\dots.$$

The detailed analysis is given in Section 2.3.1.

1.2.1.2 Average-probabilistic error

To define the average-probabilistic error of the **QS** algorithm we need to equip the input function class $\mathbb{B}_N$ with a probability measure $\mathbf{p}$. Then for $\rho\in(\frac{1}{2},1]$ we define

$$e^{\text{avg-pro}}(\mathbf{QS},\rho,\mathbf{p})=\sum_{f\in\mathbb{B}_N}\mathbf{p}(f)\min_{J:\mu(J,f)\geq\rho}\max_{j\in J}|S(f)-\mathbf{QS}(f,j)|.$$

We are then interested in the average error that holds with a certain fixed probability ρ.

In the average-probabilistic setting, we shall consider two measures on the set of Boolean functions. The first measure $\mathbf{p}_1$ is uniform on Boolean functions, while the second measure $\mathbf{p}_2$ is uniform on arithmetic means of Boolean functions. These measures have different properties. Although the mean element of the arithmetic means is $\frac{1}{2}$ for both measures, the first central moment is of order $N^{-1/2}$ for the first measure $\mathbf{p}_1$, and about $\frac{1}{4}$ for the second measure $\mathbf{p}_2$. The first central moment is exactly equal to the error of the constant algorithm that always outputs $\frac{1}{2}$. This explains why we can obtain the error of order $N^{-1/2}$ without any quantum queries for the first measure. This provides the motivation for us to check whether the error of the **QS** algorithm enjoys a similar property. Our investigation shows that that this is indeed the case iff M is divisible by 4. More precisely, the average-probabilistic error of the **QS** algorithm for the measure $\mathbf{p}_1$ satisfies

$$e^{\text{avg-pro}}(\mathbf{QS},\rho,\mathbf{p}_1)=\begin{cases}O(\min\{M^{-1},N^{-1/2}\}) & \text{for } M \text{ divisible by 4,}\\ O(M^{-1}) & \text{for } M \text{ not divisible by 4.}\end{cases} \tag{1.5}$$

Moreover, using the lower bounds of [36] when M is divisible by 4, and since we have $e^{\text{avg-pro}}(\mathbf{QS},\rho,\mathbf{p}_1)=\Omega(M^{-1})$ for M not divisible by 4 and large N/M the estimates (1.5) are essentially sharp.

For the measure $\mathbf{p}_2$, the average-probabilistic error of the **QS** algorithm is of order M^{-1} for all M. For both measures, the upper bounds mentioned earlier match lower bounds that were obtained by Papageorgiou [36]. Hence, the **QS** algorithm enjoys minimal error bounds also in the average-probabilistic setting if we choose M divisible by 4 for the first measure and with no restriction on M for the second measure.

The detailed analysis is given in Section 2.3.2.

1.2.1.3 Worst-average error

In this setting, we take the worst case performance over all Boolean functions and the average performance over all outcomes of the **QS** algorithm, so that

$$e_q^{\text{wor-avg}}(\mathbf{QS})=\sup_{N>M}\sup_{f\in\mathbb{B}_N}\left(\sum_{j=0}^{M-1}p_f(j)\,|S(f)-\mathbf{QS}(f,j)|^q\right)^{1/q}.$$

The average performance is measured in the L_q norm, $q\in[1,\infty]$. Since we do not use a classical computer to prepare the input for the **QS** algorithm we have $\bar{f}=f$ and $p_{\bar{f}}(j)=p_f(j)$. This setting is analogous to the randomized (Monte Carlo) setting used for algorithms on a classical computer. The worst-average setting seems natural for the analysis of quantum algorithms for the same reasons that motivates the worst-average-average error criterion of (1.3). The results depend on the choice of q. Obviously, for larger q, the effect of the average behavior becomes less significant. In fact, the limiting case, $q=\infty$, leads to the deterministic case (modulo sets of measure zero). Not surprisingly, for $q = \infty$, the results are negative, i.e., the error is of constant order.

We shall study error bounds for large M. Without loss of generality we assume $N > M$, so that if M tends to infinity, then so does N. To make error bounds independent of N, we take the supremum over $N > M$ in the corresponding definitions of the errors. When we speak about the sharpness of error bounds, we usually take a large M and select a still larger N and a Boolean function for which the presented error bound is sharp.

We now indicate our results for $q \in [1, \infty)$. The worst-average error $e_q^{\text{wor-avg}}(M)$ of the **QS** algorithm with M quantum queries satisfies:

$$e_q^{\text{wor-avg}}(\mathbf{QS}) = \begin{cases} \Theta(\ln M/M) & \text{for } q = 1, \\ \Theta(M^{-1/q}) & \text{for } q \in (1, \infty). \end{cases}$$

Comparing with [36], we conclude that these error bounds do not match the complexity lower bounds. We shall consider a slight modification of the **QS** algorithm to address this issue.

Quantum Boolean summation with repetitions

The error bounds of the **QS** algorithm can be improved by the use of repetitions. Namely, we repeat the **QS** algorithm $2n+1$ times and take the median of the outputs obtained as the final output. This procedure boosts the success probability of the approximation at the expense of the number of quantum queries. We show that with n independent of M and depending linearly on q, we decrease the **QS** algorithm error to be of order M^{-1}. Hence, the use of repetitions is particularly essential for large q, since we change the error bound $O(M^{-1/q})$ without repetitions to the error bound $O(M^{-1})$ with repetitions.

The error bound of order M^{-1} is optimal. This follows from the use of, for instance, Chebyshev's inequality and the fact that the lower bound $\Omega(M^{-1})$ is sharp in the worst-probabilistic setting, see [36]. Hence, the **QS** algorithm with repetitions is also optimal in the worst-average setting.

The detailed analysis of the **QS** algorithm in the worst-average setting including repetitions is given in Section 2.3.3

1.2.1.4 Quantum Boolean summation simulation

Simulating arbitrary quantum algorithms on a classical computer is very difficult due to the exponential time and memory requirements on the number of qubits. Hence, such simulations can be run only for input data of moderate size. We designed a MATLAB procedure simulating the **QS** algorithm. This simulation computes amplitudes of the **QS** algorithm final state depending on the Boolean input function and a desired accuracy ε. The results of the simulation can be used, for instance, to visualize the final state distribution. Detailed studies of the **QS** algorithm show that this task can be accomplished with cost of order $(N/\varepsilon)\log_2(1/\varepsilon)$. Here N stands for the cardinality of the domain of a Boolean function. We recall that the **QS** algorithm requires of order $\log_2 N + \log_2(1/\varepsilon)$ qubits so the memory requirement for the simulation is N/ε. Thus the cost of our simulation is optimal modulo a logarithmic factor $\log_2(1/\varepsilon)$. The presence of this factor is due to the use of the Fast Fourier Transform. Since we do not know any faster algorithm for the discrete Fourier transform, which is a part for the **QS** algorithm, this simulation achieves the best possible cost order using existing computational tools. The details are given in Section 2.4.

1.2.2 Multivariate Feynman-Kac path integration

Path integration is one of the continuous problems that has been studied studied in the quantum setting. Traub and Woźniakowski [48] studied general path integrals with smooth integrands. We consider a special case of path integrals, namely, multivariate Feynman-Kac path integrals. These path integrals arise in many problems of quantum physics, quantum chemistry, and even in financial mathematics. In particular, the multivariate Feynman-Kac path integral expresses the solution of the initial value problem for the diffusion equation. Studies of this problem in the worst-case deterministic setting are in [29, 39]. We study multivariate Feynman-Kac path integration in the randomized and quantum settings.

Path integrals are defined as integrals over an infinite dimensional space equipped with a probability measure. A path integral is called a Wiener integral if the respective measure is the Wiener measure w on the space $\mathcal{C}$ of continuous functions from $\mathbb{R}_+$ to $\mathbb{R}^d$. In the case $d \geq 2$, we add the term "multivariate". The multivariate Feynman-Kac path integral,

$$z(\mathbf{u}, t) = \int_{\mathcal{C}} v(\mathbf{x}(t) + \mathbf{u}) \exp\left(\int_0^t V(\mathbf{x}(s) + \mathbf{u})\, ds \right) w(d\mathbf{x}), \tag{1.6}$$

is the solution of the initial value problem for the multivariate diffusion equation

$$\begin{aligned} \frac{\partial z}{\partial t}(\mathbf{u}, t) &= \frac{1}{2}\Delta z(\mathbf{u}, t) + V(\mathbf{u}) z(\mathbf{u}, t) \qquad \text{for } (\mathbf{u}, t) \in \mathbb{R}^d \times (0, \infty), \\ z(\mathbf{u}, 0) &= v(\mathbf{u}). \end{aligned}$$

We assume that the input functions, initial value v and potential V belong to a certain function class F for which the integral (1.6) is well defined.

Many algorithms have been developed for the univariate case, $d = 1$, and the Feynman-Kac path integral is the solution of the diffusion equation with one space variable. Most of these algorithms are randomized. They are obtained as follows. First, the path integral is approximated by a multivariate integral over $\mathbb{R}^n$, with *large* n, and then this integral is approximately computed by using a randomized algorithm such as Monte Carlo.

A new approach was proposed in [39] for the univariate case $d = 1$. A deterministic algorithm based on L_2-approximation of v and V was constructed. This approach was modified and generalized for the multivariate case in [26, 29], and is also the basis for the randomized and quantum algorithms presented in Chapter 3.

We want to check how much the power of randomization and quantum computation helps in solving the multivariate Feynman-Kac path integration problem. This question has been addressed for general path integrals with smooth integrands in [48], where it is shown that this problem can be solved on quantum computers exponentially faster than in the worst-case deterministic setting and roughly quadratically faster than in the randomized setting. One of our questions is how the special form of the Feynman-Kac path integrals can be exploited and how it can improve the general results of [48].

1.2.2.1 Worst-case deterministic setting

Although we focus mainly on the quantum and randomized settings, we first formulate the computational problem of finding a local approximate solution of the path integral (3.3) at a given point in the worst-case deterministic setting. Assume that we have a deterministic algorithm A_n that uses n function evaluations of v and V. We define the worst-case deterministic error of the algorithm A_n as

$$e^{\mathrm{wor}}(A_n, F) = \sup_{v,V \in F} |z_{v,V}(\mathbf{u}, t) - A_n(v, V)|,$$

where $z(\mathbf{u}, t)$ in the exact solution of the diffusion equation at a given fixed point $(\mathbf{u}, t) \in \mathbb{R}^d \times (0, \infty)$.

In [26, 29] we extended the algorithm of [39] to the multivariate case. We preserved its original structure, but we used uniform approximation instead of L_2-approximation. This permits the algorithm to be used for arbitrary d, while an algorithm based on L_2-approximation, such as that of [39], can be only used for $d = 1$. The cost of computing an ε-approximation is roughly of order $\varepsilon^{-\alpha(F)}$ for a certain positive $\alpha(F)$ which depends on a given class F of input functions. The upper bound is derived from the complexity of uniform approximation for the class F. When F is a class of r times continuously differentiable d-variate functions, we have $\alpha(F) = d/r$. So in the worst case deterministic setting the problem suffers from the curse of dimensionality. Moreover, this algorithm and the one in [39] require the precomputation of a large number of coefficients which is quite difficult.

The (information) complexity of multivariate Feynman-Kac path integration in the worst-case setting is defined by

$$n^{\rm wor}(\varepsilon, F) = \min\{n : \ \exists\, A_n \quad \text{such that} \quad e^{\rm wor}(A_n) \le \varepsilon\}.$$

The complexity is bounded from below by complexity of multivariate integration, and from above by complexity of uniform approximation. Furthermore, the algorithm of [29] is almost optimal for classes F for which the complexities of the multivariate integration and the uniform approximation are of the same order. This holds, for instance, for the class F of r times continuously differentiable multivariate functions.

1.2.2.2 Randomized and quantum settings

A randomized algorithm A_n depends on a random element ω chosen from some probability space Ω, with n denoting the number of function evaluations. In the randomized setting, the computed approximation of the exact solution $z(\mathbf{u}, t)$ is then a random variable $A_n(v, V; \omega)$ that depends on a random element $\omega \in \Omega$. We measure the error of the algorithm A_n with respect to the L_2 norm, so that

$$e^{\rm rand}(A_n) = \sup_{v,V \in F} \left(\mathbb{E}(z_{v,V}(\mathbf{u}, t) - A_n(v, V; \omega))^2\right)^{1/2}. \tag{1.7}$$

We are also interested in estimating the (information) complexity $n^{\rm rand}(\varepsilon, F)$ of multivariate Feynman-Kac path integration, i.e., the minimal expected number of functions v and V values needed to compute an ε-approximation in the randomized setting, which is given by

$$n^{\rm rand}(\varepsilon, F) = \min\{n : \ \exists\, A_n \quad \text{such that} \quad e^{\rm rand}(A_n) \le \varepsilon\}.$$

We now turn to the quantum setting. We met some technical difficulties dealing with deterministic (bit) queries, so we left this case as one of the open problems in Chapter 4. Our results are only established for randomized queries.

We now outline the results obtained. We consider quantum algorithms with randomized queries. The error of a quantum algorithm A_n is defined analogously to

Section 1.1.1 as

$$e^{\mathrm{quant}}(A_n,P_s)=\sup_{v,V\in F}\left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}\,|z_{v,V}(\mathbf{u},t)-A_{n,\omega}(P_s(v,V),j;\omega)|^2\right)^{1/2}, \qquad (1.8)$$

where $\mathbb{E}$ is the expectation over the probability space Ω, and $\mathbb{E}_{\mathrm{q}}$ is the expectation with respect to the distribution of the quantum algorithm outcomes. We recall that P_s is a classical algorithm using s function evaluations that prepares the input for the quantum algorithm, see Section 1.1.1. Similarly to the randomized setting, we also want to know the minimal number

$$n^{\mathrm{quant}}(\varepsilon,H)=\min\{s+n:\ \exists\,P_s\ \exists\,A_n\quad\text{such that}\quad e^{\mathrm{quant}}(A_n,P_s)\le\varepsilon\}.$$

of randomized quantum queries and classical function evaluations, which are needed to guarantee that the error does not exceed ε.

We present algorithms that compute an ε-approximation, i.e., with the errors (1.7) and (1.8) at most ε, and provide their cost analyses. These algorithms are also based on uniform approximation. However, the power of randomization and quantum computation permits us to improve the worst-case deterministic complexity bound $O(\varepsilon^{-\alpha(F)})$. The number of function evaluations required by the randomized algorithm is roughly of order $\varepsilon^{-2\alpha(F)/(\alpha(F)+2)}$, and the number of function evaluations and queries required by the quantum algorithm is roughly of order $\varepsilon^{-\alpha(F)/(\alpha(F)+1)}$. We stress that the exponent of ε^{-1} in the randomized setting is at most 2, and in the quantum setting is at most 1.

We also study the complexity of multivariate Feynman-Kac path integration in the randomized and quantum settings. The complexity is bounded from below by the complexity of multivariate weighted integration, just as in [39]. The upper bounds are provided by the costs of the algorithms mentioned above.

For the class F of r times continuously differentiable d-variate functions we have $\alpha(F)=d/r$. In the randomized setting, the complexity is roughly $\Theta(\varepsilon^{-2/(1+2r/d)})$; in the quantum setting it is roughly $\Theta(\varepsilon^{-1/(1+r/d)})$. Furthermore, we know algorithms that use $O(\varepsilon^{-2})$ function values in the randomized setting, and $O(\varepsilon^{-1})$ function values in the quantum setting, with the factors in the big O notation independent of the number of variables d. In both cases, the curse of dimensionality is vanquished. We thus have exponential speedup over the worst case setting. For $d \gg r$, we have quadratic speedup of the complexity in the quantum setting over the complexity in the randomized setting. We refer the reader to Chapter 3 for a detailed discussion.

Chapter 2

Quantum Boolean summation

2.1 Introduction

The quantum summation (**QS**) algorithm (also known as the amplitude estimation algorithm) of Brassard, Høyer, Mosca and Tapp [4] computes an approximation to the arithmetic mean of all values of a Boolean function defined on a set of $N = 2^n$ elements. We denote the class of such functions by $\mathbb{B}_N$. Information regarding the Boolean function is supplied by quantum queries. Quantum queries play a role similar to that of function evaluations in the classical worst-case and randomized settings. Suppose that we use $M - 1$ quantum queries. The only case of interest is when M is much smaller than N. It was proved in [4] that the error of the **QS** algorithm is at most

$$\frac{\pi}{M} + \frac{\pi^2}{M^2} \quad \text{with probability } \frac{8}{\pi^2} = 0.81\ldots. \tag{2.1}$$

Nayak and Wu [31] showed that for any $p \in (\frac{1}{2}, 1)$ the error of any quantum algorithm that uses no more than $M - 1$ quantum queries must be proportional to M^{-1} with probability p. Therefore, the **QS** algorithm enjoys the smallest possible error modulo a factor multiplying M^{-1}.

The minimal error estimate of order M^{-1} in the quantum setting should be compared to the minimal error estimates in the worst case and randomized settings of algorithms using $M-1$ function values. It is known, see [34], that in the worst case setting, the error bound is roughly $\frac{1}{2}(1-M/N)$. This means that, as long as M is much less than N, the error is almost $\frac{1}{2}$, and is therefore $O(M)$ times larger than that in the quantum setting. In the randomized setting, the classical Monte Carlo is almost optimal, and the error bound is roughly $1/(2\sqrt{M})$, see again [34]. Hence, the error $O(\sqrt{M})$ larger than that in the quantum setting.

We check whether the estimate (2.1) is sharp and how the error decreases if we decrease the probability $p = 8/\pi^2$ to $p > \frac{1}{2}$. We also study the error of the **QS** algorithm in various settings. The estimate (2.1) corresponds to the worst-probabilistic setting, which is most frequently used for quantum algorithms. The average performance of the **QS** algorithm with respect to a measure on Boolean functions is studied in the average-probabilistic setting. In the worst-average setting, we study the worst performance with respect to Boolean functions and the average performance with respect to all outcomes of a quantum algorithm. This setting is usually used for the study of the classical Monte Carlo algorithm. We study error bounds in the worst- and average-probabilistic settings with probabilities $p \in (1/2, 8/\pi^2]$. If we want to obtain error bounds with higher probability it is enough to run the **QS** algorithm a number of times and take the median as the final result, see e.g., [15, 32].

In the worst-probabilistic setting, we show that (2.1) can be slightly improved. Namely, the error of the **QS** algorithm is at most

$$\frac{3}{4}\,\frac{\pi}{M} \quad \text{with probability } \frac{8}{\pi^2}.$$

Furthermore, for large M and N/M we prove that the last estimate is sharp. In particular, for $p \in (1/2, 8/\pi^2]$ we prove that the error of the **QS** algorithm is at most

$$\frac{\bigl(1 - v^{-1}(p)\bigr)\,\pi}{M} \qquad \text{with probability}\, p,$$

where v^{-1} is the inverse of the function $v(\Delta) = \sin^2(\pi\Delta)/(\pi\Delta)^2$. We have $1-v^{-1}(p) \in (\frac{1}{2}, \frac{3}{4}]$ and it is well approximated by $\frac{1}{16}\pi^2 p + \frac{1}{4}$. For the frequently considered values of p we have

$$\big(1 - v^{-1}(\tfrac{1}{2}+)\big)\pi = 1.75\ldots, \quad \big(1 - v^{-1}(\tfrac{3}{4})\big)\pi = 2.23\ldots,$$
$$\big(1 - v^{-1}(8/\pi^2)\big)\pi = \tfrac{3}{4}\pi = 2.35\ldots.$$

In the average-probabilistic setting, we consider two measures on the set of Boolean functions. The first measure is uniform on Boolean functions, while the second measure is uniform on arithmetic means of Boolean functions. The results for these two measures are quite different. The mean element is $\frac{1}{2}$ for both measures. However, the first moment is of order $N^{-1/2}$ for the first measure, and about $\frac{1}{4}$ for the second. The first moment is exactly equal to the error of the constant algorithm that always outputs $\frac{1}{2}$, which explains why we can obtain the error of order $N^{-1/2}$ without any quantum queries for the first measure. This provides the motivation for us to check whether the error of the **QS** algorithm enjoys a similar property. It turns out that this is indeed the case iff M is divisible by 4. That is, for M divisible by 4, the average-probabilistic error of the **QS** algorithm is of order $\min\{M^{-1}, N^{-1/2}\}$, and if M is not divisible by 4, then the error is of order M^{-1}. For the second measure, since the first moment is not small, the average-probabilistic error of the **QS** algorithm is of order M^{-1} for all M. For both measures, the upper bounds presented in this paper match lower bounds from [36]. Hence, the **QS** algorithm enjoys minimal error bounds also in the average-probabilistic setting if we choose M divisible by 4 for the first measure and with no restriction on M for the second measure.

In the worst-average setting, we take the worst case performance over all Boolean functions and the average performance over all outcomes of the **QS** algorithm. The average performance is measured in the L_q norm, where $q \in [1, \infty]$. This setting is analogous to the randomized (Monte Carlo) setting used for algorithms on a classical computer. Recall that, for a number of reasons discussed in Section 1.2.1.3, the worst-average setting is a natural choice for the analysis of quantum algorithms. As we shall see, the estimates depend on the choice of q. Obviously, for larger q, the effect of the average behavior becomes less significant. In fact, the limiting case, $q = \infty$, leads to the deterministic case (modulo sets of measure zero). Not surprisingly, the results are negative for $q = \infty$. In what follows, we indicate error bounds for large M. Since we always assume that $M < N$, this means that for M tending to infinity we also let N tend to infinity. To make error bounds independent of N, we take the supremum over $N > M$ in the corresponding definitions of the errors. When we speak about the sharpness of error bounds, we usually take a large M and select a still larger N and a Boolean function for which the presented error bound is sharp. The worst-average error of the **QS** algorithm with M quantum queries satisfies the following:

- For $q = 1$, the worst-average error is $\Theta(\ln M/M)$. Furthermore, the asymptotic constant is $2/\pi$ for $M - 2$ divisible by 4.

- For $q \in (1, \infty)$, the worst-average error is $\Theta\big(M^{-1/q}\big)$. Furthermore, the asymptotic constant is approximately $\big(\int_0^\pi \sin^{q-2}(x)dx/\pi\big)^{1/q}$ for $M - 2$ divisible by 4 and q close to 1.

- For $q = \infty$, the worst-average error is constant and equals 1.

The error bounds of the **QS** algorithm are improved by the use of repetitions. Namely, we repeat the **QS** algorithm $2n+1$ times and take the median of the outputs obtained as the final output. This procedure boosts the success probability of the approximation at the expense of the number of quantum queries. We show that with n independent of M and depending linearly on q, we decrease the **QS** algorithm error to be of order M^{-1}. Hence, the use of repetitions is particularly important for large q since we change the error bound $O(M^{-1/q})$ without repetitions to the error bound $O(M^{-1})$ with repetitions. The constant in the last big O notation is absolute and does not depend on q.

The error bound of order M^{-1} is optimal. This follows from the use of, for instance, Chebyshev's inequality and the fact that the lower bound $\Omega(M^{-1})$ is sharp in the worst-probabilistic setting, see also [36]. Hence, the **QS** algorithm with repetitions is optimal in the worst-average setting.

We now outline the contents of this chapter. In Section 2.2 we define the **QS** algorithm. In Section 2.3 we define precisely the error settings discussed above and analyze the performance of the **QS** algorithm in these error settings.

2.2 Quantum summation algorithm

We consider the most basic form of the summation problem, i.e., the summation of Boolean functions. Let $\mathbb{B}_N$ denote the set of Boolean functions $f : \{0, \ldots, N-1\} \to \{0, 1\}$. Let

$$a_f = \frac{1}{N} \sum_{i=0}^{N-1} f(i)$$

denote the arithmetic mean of all values of f. Clearly, $a_f \in [0, 1]$.

Problem. *For $f \in \mathbb{B}_N$, compute an ε-approximation $\bar{a}_f$ of the sum a_f such that*

$$|\bar{a}_f - a_f| \le \varepsilon. \tag{2.2}$$

We are interested in the minimal number of evaluations of the function f that is needed to compute $\bar{a}_f$ satisfying (2.2). It is known that in the worst case setting, we need roughly $N(1-\varepsilon)$ evaluations of the function f. In the randomized setting, we assume that $\bar{a}_f$ is a random variable. We replace (2.2) by the requirement that the expected value of $|\bar{a}_f - a_f|$ is at most ε for any function f. It is known, see e.g., [34], that in the randomized setting we need roughly $\min\{N, \varepsilon^{-1/2}\}$ function evaluations. In the quantum setting, we want to compute a random variable $\bar{a}_f$ such that (2.2) holds with high probability (greater than $\frac{1}{2}$), either for *all* Boolean functions or on the average with respect to a probability measure defined on the set $\mathbb{B}_N$. These two error criteria in the quantum setting will be precisely defined in Section 2.3.

In this section we describe the quantum summation algorithm, which is also called the quantum amplitude estimation algorithm. This algorithm was discovered by Brassard, Høyer, Mosca and Tapp [4], and uses Grover's iterate operator as its basic component, see [13]. We use standard notation of quantum computation, see e.g., [32].

For simplicity we assume that $N = 2^n$. Let $\mathcal{H}_n$ denote the tensor product $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ of n copies of $\mathbb{C}^2$, with $\mathbb{C}^2$ the 2-dimensional complex vector space. Unit vectors from $\mathbb{C}^2$ are called *one qubit* quantum states (or *qubits*). Let $|0\rangle$ and $|1\rangle$ be an orthonormal basis of $\mathbb{C}^2$. Then any qubit $|\psi\rangle$ can be represented as

$$|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle \qquad \text{with} \quad \psi_k \in \mathbb{C} \quad \text{and} \quad |\psi_0|^2 + |\psi_1|^2 = 1.$$

For $j = 0, 1, \dots, N-1$, we have $j = \sum_{k=0}^{n-1} 2^{n-1-k} j_k$, with $j_k \in \{0, 1\}$. Let

$$|j\rangle = \bigotimes_{k=0}^{n-1} |j_k\rangle.$$

The set $\{|j\rangle : j = 0, \dots, N-1\}$ forms an orthonormal basis of $\mathcal{H}_n$ and any unit vector $|\psi\rangle \in \mathcal{H}_n$ can be represented as

$$|\psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle \qquad \text{with} \quad \psi_j \in \mathbb{C} \quad \text{and} \quad \sum_{j=0}^{N-1} |\psi_j|^2 = 1.$$

Unit vectors from $\mathcal{H}_n$ are called n qubit quantum states (or quantum states or just states, whenever n is clear from the context).

The only transformations that can be performed on quantum states are defined by certain unitary operators on $\mathcal{H}_n$. We now define the six unitary operators that are basic components of the summation algorithm. Since unitary operators are linear, it is enough to define them on the basis states $|j\rangle$.

1. Let $S_0 : \mathcal{H}_n \to \mathcal{H}_n$ denote the *inversion about zero transform*

   $$S_0|j\rangle = (-1)^{\delta_{j,0}}|j\rangle,$$

   where $\delta_{j,0}$ is the Kronecker delta. Hence, $S_0|0\rangle = -|0\rangle$ and $S_0|j\rangle = |j\rangle$ for all $j \neq 0$. This corresponds to the diagonal matrix with one element equal to -1, and the rest equal to 1. We claim that the operator S_0 can be also written as the Householder operator

$$S_0 = I - 2|0\rangle\langle 0|.$$

Here, for a state $|\psi\rangle$, we let $|\psi\rangle\langle\psi|$ denote the projection onto the space $\mathrm{span}\{|\psi\rangle\}$ given by

$$(|\psi\rangle\langle\psi|)\,|x\rangle = \langle\psi|x\rangle\,|\psi\rangle,$$

where $\langle\psi|x\rangle$ is the inner product in $\mathcal{H}_n$, $\langle\psi|x\rangle = \sum_{k=0}^{N-1} \overline{\psi_k} x_k$. The matrix form of the projector $|\psi\rangle\langle\psi|$ in the basis $\{|j\rangle\}$ is $(\overline{\psi_k}\psi_j)_{j,k=0}^{N-1}$. One can also view the matrix form of the projector $|\psi\rangle\langle\psi|$ as the matrix product of the $N \times 1$ column vector $|\psi\rangle$ and the $N \times 1$ row vector $\langle\psi| = |\psi\rangle^\dagger$, which is the Hermitian conjugate of $|\psi\rangle$. To prove this claim, note that for any $|x\rangle = \sum_{j=0}^{N-1} x_j|j\rangle \in \mathcal{H}_n$ we have

$$\langle k|\big(I - 2|0\rangle\langle 0|\big)|x\rangle = \langle k|x\rangle - 2\langle 0|x\rangle\langle k|0\rangle = \begin{cases} x_k - 2x_k = -x_k & \text{for } k = 0, \\ x_k - 0 = x_k & \text{for } k \neq 0. \end{cases}$$

Hence, $I - 2|0\rangle\langle 0| = S_0$, as claimed.

2. Let $W_N : \mathcal{H}_n \to \mathcal{H}_n$ denote the *Walsh-Hadamard transform*

$$W_N|j\rangle = \frac{1}{\sqrt{N}} \bigotimes_{k=0}^{n-1} \big(|0\rangle + (-1)^{j_k}|1\rangle\big).$$

That is, the Walsh-Hadamard transform corresponds to the matrix with entries

$$\begin{aligned} \langle i|W_N|j\rangle = \frac{1}{\sqrt{N}} \prod_{k=0}^{n-1} \langle i_k| \big(|0\rangle + (-1)^{j_k}|1\rangle\big) \\ = \frac{1}{\sqrt{N}} \prod_{k=0}^{n-1} (-1)^{i_k j_k} = \frac{1}{\sqrt{N}} (-1)^{\sum_{k=0}^{n-1} i_k j_k}. \end{aligned}$$

The matrix $(\langle i|W_N|j\rangle)_{i,j=0}^{N-1}$ is symmetric. Furthermore,

$$\begin{aligned} W_N^2|j\rangle &= \frac{1}{\sqrt{N}} W_n \bigotimes_{k=0}^{n-1} \big(|0\rangle + (-1)^{j_k}|1\rangle\big) \\ &= \frac{1}{\sqrt{N}} \bigotimes_{k=0}^{n-1} \left(\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) + \frac{(-1)^{j_k}}{\sqrt{2}} (|0\rangle - |1\rangle) \right) \\ &= \frac{1}{\sqrt{N}} \bigotimes_{k=0}^{n-1} \sqrt{2}|j_k\rangle \ = \ |j\rangle. \end{aligned}$$

Thus, $W_N^2 = I$ and $W_N^{-1} = W_N$ is orthogonal. This means that the operator W_N is symmetric and unitary.

3. For $K = 1, 2, \ldots, 2^n$, let $F_{K,n} : \mathcal{H}_n \to \mathcal{H}_n$ denote the *quantum Fourier transform*

$$F_{K,n}|j\rangle = \begin{cases} K^{-1/2} \sum_{k=0}^{K-1} e^{2\pi ijk/K}|k\rangle, & \text{for } j = 0, 1, \ldots, K-1, \quad (i = \sqrt{-1}) \\ |j\rangle & \text{for } j = K, \ldots, 2^n - 1. \end{cases}$$

Hence, $F_{K,n}$ corresponds to the unitary block-diagonal matrix

$$\begin{bmatrix} F_K & 0 \\ 0 & I \end{bmatrix},$$

where $F_K = \left(K^{-1/2}\, e^{2\pi ijk/k}\right)_{j,k=0}^{K-1}$ is the matrix of the quantum Fourier transform. For $K = 2^n = N$ we have

$$F_{N,n}|\psi\rangle = \sum_{j=0}^{N-1} \psi_j F_{N,n}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} \psi_j e^{2\pi ijk/N} \right) |k\rangle.$$

The coefficients of $F_{N,n}|\psi\rangle$ in the basis $\{|j\rangle\}$ are the quantum Fourier transforms of the coefficients of the state $|\psi\rangle$. Note that W_N and $F_{N,n}$ coincide for the state $|0\rangle$, i.e.,

$$W_N\,|0\rangle \ = \ F_{N,n}\,|0\rangle \ = \ \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle.$$

4. Let $S_f : \mathcal{H}_n \to \mathcal{H}_n$ denote the *quantum query* operator

$$S_f|j\rangle = (-1)^{f(j)}|j\rangle,$$

This again corresponds to the diagonal matrix with elements ±1 depending on the values of the Boolean function f. This operator is the only one that provides information about the Boolean function f. This is analogous to the concept of an *oracle* or a *black-box* which is used in classical computation and which supplies information about the function f through its values.

The standard definition of the quantum query $\bar{S}_f$ is

$$\bar{S}_f : \mathcal{H}_n \otimes \mathbb{C}^2 \to \mathcal{H}_n \otimes \mathbb{C}^2, \qquad \bar{S}_f|j\rangle|i\rangle = |j\rangle|i \oplus f(j)\rangle,$$

where $\oplus$ means addition modulo 2. We can simulate S_f by $\bar{S}_f$ if we use an auxiliary qubit $(1/\sqrt{2})(|1\rangle - |0\rangle)$, namely,

$$\begin{aligned} \bar{S}_f\left(|j\rangle\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right) &= |j\rangle\frac{|1\oplus f(j)\rangle - |f(j)\rangle}{\sqrt{2}} \\ &= (-1)^{f(j)}|j\rangle\frac{|1\rangle - |0\rangle}{\sqrt{2}} = \big(S_f|j\rangle\big)\frac{|1\rangle - |0\rangle}{\sqrt{2}}. \end{aligned}$$

5. Let $Q_f : \mathcal{H}_n \to \mathcal{H}_n$ denote the *Grover operator*

$$Q_f = -W_N\, S_0\, W_N^{-1}\, S_f.$$

This is the basic component of Grover's search algorithm, see [13]. As we shall see, Q_f also plays a major role for the summation algorithm. The eigenvectors and eigenvalues of Q_f will be useful in further considerations. Let

$$|\psi\rangle = W_N|0\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|k\rangle$$

and $|\psi_0\rangle$, $|\psi_1\rangle$ denote the orthogonal projections of $|\psi\rangle$ onto the subspaces $\mathrm{span}\{|j\rangle : f(j) = 0\}$ and $\mathrm{span}\{|j\rangle : f(j) = 1\}$, respectively. That is,

$$|\psi_j\rangle = \frac{1}{\sqrt{N}}\sum_{k:\, f(k)=j}|k\rangle \qquad j = 0, 1.$$

Then $|\psi\rangle = |\psi_0\rangle + |\psi_1\rangle$ and $\langle\psi_0|\psi_1\rangle = 0$. Furthermore, $\langle\psi_j|\psi_j\rangle = N^{-1}\sum_{k:\,f(k)=j} 1$, for $j = 0,1$, so that $\langle\psi_1|\psi_1\rangle = a$ and $\langle\psi_0|\psi_0\rangle = 1 - a$, where $a = a_f$ is the sum we want to approximate.

From [4], we know that

$$\begin{aligned} Q_f|\psi_0\rangle &= (1-2a)|\psi_0\rangle + 2(1-a)|\psi_1\rangle, \\ Q_f|\psi_1\rangle &= -2a|\psi_0\rangle + (1-2a)|\psi_1\rangle. \end{aligned} \tag{2.3}$$

For the sake of completeness, we provide a short proof of (2.3). By the definition of the operator S_f we have

$$S_f|\psi_j\rangle = (-1)^j|\psi_j\rangle, \quad j = 0,1,$$

and

$$W_N\,S_0\,W_N^{-1} \;=\; W_N(I - 2|0\rangle\langle 0|)\,W_N^{-1} \;=\; I - 2(W_N|0\rangle\langle 0|W_N).$$

Since $\langle 0|W_N = (W_N|0\rangle)^\dagger = (|\psi\rangle)^\dagger = \langle\psi|$, we obtain

$$\begin{aligned} W_N\,S_0\,W_N^{-1}|\psi_j\rangle &= |\psi_j\rangle - 2(|\psi\rangle\langle\psi|)\,|\psi_j\rangle \\ &= |\psi_j\rangle - 2\langle\psi|\psi_j\rangle|\psi\rangle = |\psi_j\rangle - 2\langle\psi_j|\psi_j\rangle|\psi\rangle. \end{aligned}$$

for $j = 0,1$. From this we calculate

$$\begin{aligned} Q_f|\psi_j\rangle &= (-1)^{1+j}W_N S_0 W_N^{-1}|\psi_j\rangle \\ &= (-1)^{\delta_{j,0}}\left(|\psi_j\rangle - 2(\delta_{j,1}a + \delta_{j,0}(1-a))(|\psi_0\rangle + |\psi_1\rangle)\right), \end{aligned}$$

for $j = 0,1$, which is equivalent to (2.3).

Thus, the space $\operatorname{span}\{|\psi_0\rangle, |\psi_1\rangle\}$ is an invariant space of Q_f and its eigenvectors and corresponding eigenvalues can be computed by solving the eigenproblem for the 2×2 matrix

$$\begin{bmatrix} 1-2a & -2a \\ 2(1-a) & 1-2a \end{bmatrix}.$$

If $a \in (0,1)$, then the eigenvalues of Q_f are

$$\lambda_\pm = 1 - 2a \pm 2i\sqrt{a(1-a)} = e^{\pm 2i\theta_a}, \qquad \theta_a = \arcsin\sqrt{a},$$

and the corresponding orthonormalized eigenvectors are

$$|\psi_\pm\rangle = \frac{1}{\sqrt{2}}\left(\pm\frac{i}{\sqrt{1-a}}|\psi_0\rangle + \frac{1}{\sqrt{a}}|\psi_1\rangle\right).$$

Moreover, it is easy to check that

$$|\psi\rangle = \frac{-i}{\sqrt{2}}\left(e^{i\theta_a}|\psi_+\rangle - e^{-i\theta_a}|\psi_-\rangle\right). \tag{2.4}$$

If $a \in \{0,1\}$, then we have $\mathrm{span}\{|\psi_0\rangle, |\psi_1\rangle\} = \mathrm{span}\{|\psi\rangle\}$ and $|\psi\rangle$ is the eigenvector of Q_f with eigenvalues ± 1, respectively. For $a \in \{0,1\}$, we define

$$|\psi_+\rangle \;=\; i^{1-a}\sqrt{2}\,|\psi\rangle \quad \text{and} \quad |\psi_-\rangle = 0.$$

Then it is easy to check that (2.4) is valid, and $\lambda_\pm = e^{\pm 2i\theta_a} = (-1)^a$ is an eigenvalue of Q_f for all $a \in [0,1]$.

6. The next unitary transform, called the *Grover iterate operator*, is defined on the tensor product of $\mathcal{H}_m \otimes \mathcal{H}_n$ and uses $m+n$ qubits. The number m is related to the accuracy of the quantum summation algorithm, whereas n is related to the size of the problem. The *Grover iterate* operator $\Lambda_m(Q_f) : \mathcal{H}_m \otimes \mathcal{H}_n \to \mathcal{H}_m \otimes \mathcal{H}_n$ is defined by

   $$\Lambda_m(Q_f)\,|j\rangle|y\rangle = |j\rangle\, Q_f^j|y\rangle \quad \text{for} \quad |j\rangle|y\rangle \in \mathcal{H}_m \otimes \mathcal{H}_n.$$

   Hence, the power of Q_f applied to the second component depends on the first one. Note that j may vary from 0 to $2^m - 1$. Therefore $\Lambda_m(Q_f)$ may use the powers of Q_f up to the $(2^m - 1)$st.

We need one more concept of quantum computation, that of *measurement.* Suppose s is a positive integer and consider the space $\mathcal{H}_s$. Given the state

$$|\psi\rangle = \sum_{k=0}^{2^s-1} \psi_k|k\rangle \;\in\; \mathcal{H}_s,$$

we cannot, in general, recover all the coefficients ψ_k. We can only measure the state $|\psi\rangle$ with respect to a finite collection of linear operators $\{M_j\}_{j=0}^{p}$, where the $M_j : \mathcal{H}_s \to \mathcal{H}_s$ satisfy the completeness relation

$$\sum_{j=0}^{p} M_j^\dagger M_j = I.$$

After performing the measurement, we obtain the outcome j, and the state $|\psi\rangle$ collapses into the state

$$\frac{1}{\sqrt{\langle\psi|M_j^\dagger M_j|\psi\rangle}}\, M_j|\psi\rangle;$$

this occur with probability $\langle\psi|M_j^\dagger M_j|\psi\rangle$. Note that for $M_j|\psi\rangle = 0$ the outcome j cannot occur with positive probability. Hence, with probability 1 the outcome j corresponds to $M_j|\psi\rangle \neq 0$.

The most important example of such a collection of operators is $\{|j\rangle\langle j|\}_{j=0}^{2^s-1}$. Then, the measurement of the state $|\psi\rangle$ with respect to this collection of operators gives us the outcome j and the state $|\psi\rangle$ collapses into the state

$$\frac{\langle j|\psi\rangle}{|\langle j|\psi\rangle|}|j\rangle$$

with probability $|\psi_j|^2$, $j = 0, 1, \ldots, 2^s - 1$.

Another example is a variation of the previous example and will be used in the quantum summation algorithm. We now let $s = m + n$, as for the Grover iterate operator, and define $M_j : \mathcal{H}_m \otimes \mathcal{H}_n \to \mathcal{H}_m \otimes \mathcal{H}_n$ by

$$M_j \;=\; |j\rangle\langle j| \otimes I$$

for $j = 0, 1, \ldots, 2^m - 1$, with I denoting the identity operator on $\mathcal{H}_n$. That is,

$$(|j\rangle\langle j| \otimes I)\ |x\rangle|y\rangle \;=\; \langle j|x\rangle\, |j\rangle|y\rangle$$

for $|x\rangle \in \mathcal{H}_m$ and $|y\rangle \in \mathcal{H}_n$.

Since $\sum_{j=0}^{2^m-1} (|j\rangle\langle j| \otimes I)\, |x\rangle|y\rangle = |x\rangle|y\rangle$ for all basis states $|x\rangle$ of $\mathcal{H}_m$ and $|y\rangle$ of $\mathcal{H}_n$, the completeness relation is satisfied. Consider now the probability of the outcome j

for a special state $|\psi\rangle$ of the form $|\psi\rangle = |\psi_1\rangle|\psi_2\rangle$ with $|\psi_1\rangle \in \mathcal{H}_m$, $|\psi_2\rangle \in \mathcal{H}_n$, where $\langle\psi_k|\psi_k\rangle = 1$ for $k = 1, 2$. Since $|j\rangle\langle j| \otimes I$ is self-adjoint, the outcome j and the collapse of the state $|\psi\rangle$ to the state

$$\frac{\langle j|\psi_1\rangle}{|\langle j|\psi_1\rangle|}\,|j\rangle|\psi_2\rangle$$

occur with probability $|\langle j|\psi_1\rangle|^2$. Hence, this collection of operators measures the components of the so-called first register $|\psi_1\rangle$ of the quantum state $|\psi\rangle$.

Following [4], we are ready to describe the quantum summation (**QS**) algorithm for solving our problem. The **QS** algorithm depends on a Boolean function f and on an integer parameter M that controls the number of quantum queries and the accuracy of the algorithm. We perform computations in the space $\mathcal{H}_m \otimes \mathcal{H}_n$, with $m = \lceil \log_2 M \rceil$, so we use $m + n$ qubits. As we will see later, the accuracy of the algorithm is related to the dimension of the space $\mathcal{H}_m$.

Algorithm QS$(f,\, M)$

Input state: $|0\rangle|0\rangle \in \mathcal{H}_m \otimes \mathcal{H}_n$ with $m = \lceil \log_2 M \rceil$ and $n = \log_2 N$.

Computation:

1. $|\eta_1\rangle = F_{M,m} \otimes W_N\, |0\rangle|0\rangle$,
2. $|\eta_2\rangle = \Lambda_m(Q_f)\, |\eta_1\rangle$,
3. $|\eta_3\rangle = (F_{M,m}^{-1} \otimes I)\, |\eta_2\rangle$.

Measurement:

Perform the measurement of the state $|\eta_3\rangle$ with respect to the collection $\{\, (|j\rangle\langle j|) \otimes I\, \}_{j=0}^{2^m-1}$. Denote the outcome by j.

Output: $\bar{a}_f(j) = \sin^2\big(\pi j/M\big)$.

We briefly comment on the **QS** algorithm. The input state is always the same and does not depend on f. Step 1 computes $|\eta_1\rangle = (NM)^{-1/2}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1}|j\rangle|k\rangle$, which is the equally weighted superposition of the basis states. Step 2 computes $|\eta_2\rangle$ by using the Grover iterate operator. During this step, we use the successive powers of the Grover operator Q_f; this is the only step where information about the Boolean function f is used. We shall see that the **QS** algorithm uses $M-1$ quantum queries. Step 3 computes $|\eta_3\rangle$ by performing the inverse quantum Fourier transform on the first m qubits, and prepares the system for measurement. After Step 3, we perform the measurement, obtain the outcome j and compute the output $\bar{a}_f(j)$ on a classical computer. We stress that the distribution of the outcomes j depends on the Boolean function f, and this is the only dependence of the output $\bar{a}_f(j)$ on f.

2.3 Performance analysis

In this section we analyze the error of the **QS** algorithm. As we have seen in Section 2.2, the output $\bar{a}_f(j)$ of the **QS** algorithm is a random value chosen according to a certain distribution that depends on the input function f. In this sense, the **QS** algorithm is a randomized algorithm. Various ways of measuring the performance of randomized algorithms are commonly used in the analysis of algorithms and computational complexity; these all correspond to various error criteria. In particular, we consider three error criteria: worst-probabilistic, average-probabilistic and worst-average.

Worst-probabilistic error

We start with the error criterion that is used in most papers dealing with quantum computation. We are interested in the worst case error of the **QS** algorithm that holds with a given probability p. Here $p \in [0,1]$ and $1-p$ measures the probability of **QS** algorithm's failure and usually p is set to be $\frac{3}{4}$. In our analysis, however, we will allow an arbitrary $p \in (1/2, 8/\pi^2]$. The choice of the upper bound $8/\pi^2 = 0.81\ldots$ will be clear from the analysis of the **QS** algorithm. The **QS** algorithm outputs $\bar{a}_f(j)$,

$j = 0, 1, \ldots, M-1$, with probability $p_f(j)$, which is given in (2.6) of Theorem 2.3.2. The worst-probabilistic error of **QS** is formally defined as the smallest error bound that holds for all Boolean functions with probability at least p, i.e.,

$$e^{\text{wor-pro}}(M,p) = \inf \left\{ \alpha : \sum_{j\colon |a_f - \bar{a}_f(j)| \le \alpha} p_f(j) \ge p \qquad \forall f \in \mathbb{B}_N \right\}.$$

It is easy to see that $e^{\text{wor-pro}}(M,p)$ can be rewritten as follows. Let $J \subset \{0, 1, \ldots, M-1\}$. For $f \in \mathbb{B}_N$ define the measure of J as

$$\mu(J,f) = \sum_{j \in J} p_f(j).$$

Then

$$e^{\text{wor-pro}}(M,p) = \max_{f \in \mathbb{B}_N} \min_{J\colon \mu(J,f) \ge p} \max_{j \in J} |a_f - \bar{a}_f(j)|. \tag{2.5}$$

Average-probabilistic error

The worst-probabilistic error $e^{\text{wor-pro}}(M,p)$ of the **QS** algorithm is defined by the worst performance with respect to Boolean functions. It is also natural to consider the average performance of the **QS** algorithm with respect to Boolean functions. Let $\mathbf{p}$ be a probability measure on the set $\mathbb{B}_N$, so that each Boolean function $f \in \mathbb{B}_N$ occurs with probability $\mathbf{p}(f)$. Obviously, $\mathbf{p}(f) \ge 0$ and $\sum_{f \in \mathbb{B}_N} \mathbf{p}(f) = 1$. The average-probabilistic error is defined by replacing the first max in (2.5) by the expectation, i.e.,

$$e^{\text{avg-pro}}(M,p) = \sum_{f \in \mathbb{B}_N} \mathbf{p}(f) \min_{J\colon \mu(J,f) \ge p} \max_{j \in J} |a_f - \bar{a}_f(j)|,$$

Hence, we are interested in the average error that holds with a certain fixed probability.

Worst-average error

The worst-average error corresponds to the worst case performance with respect to all Boolean functions from $\mathbb{B}_N$ and the average performance with respect to all outcomes.

This average performance is measured by the expectation in the L_q norm, $q \in [1, \infty]$, with respect to the probability measure of all outcomes provided by the **QS** algorithm. As mentioned before, we make the worst-average error independent of N by taking the supremum over $N > M$. That is, the worst-average error is defined as:

- for $q \in [1, \infty)$,

$$e_q^{\text{wor-avg}}(M) = \sup_{N>M} \max_{f \in \mathbb{B}_N} \left(\sum_{j=0}^{M-1} p_f(j) \, |a_f - \bar{a}_f(j)|^q \right)^{1/q},$$

- for $q = \infty$,

$$e_\infty^{\text{wor-avg}}(M) = \sup_{N>M} \max_{f \in \mathbb{B}_N} \max_{j:\, p_f(j)>0} |a_f - \bar{a}_f(j)|.$$

It is easy to check that for $q = \infty$, the **QS** algorithm behaves badly. Indeed, if M is odd, we can take f with all values one, and then $a_f = 1$, $p_f(0) = 1/M^2$ and $\bar{a}_f(0) = 0$. Hence $e_\infty^{\text{wor-avg}}(M) = 1$. If M is even, we take f with only one value equal to 1, and then $a_f = 1/N$, $p_f(M/2) > 0$ and $\bar{a}_f(M/2) = 1$. Hence, $|a_f - \bar{a}_f(M/2)| = 1 - 1/N$ and $e_\infty^{\text{wor-avg}}(M) = 1$.

Therefore, in the rest of the chapter, we consider only $q \in [1, \infty)$. As we shall see, the cases $q > 1$ and $q = 1$ will require different analyses and lead to quite different results.

2.3.1 Worst-probabilistic error

We begin by citing a theorem from [4] for which we shall propose a number of improvements.

Theorem 2.3.1. *For any Boolean function $f \in \mathbb{B}_N$, the* **QS** *algorithm uses exactly $M - 1$ quantum queries and outputs $\bar{a}$ that approximates $a = a_f$ such that*

$$|\bar{a} - a| \le \frac{2\pi}{M} \sqrt{a(1-a)} + \frac{\pi^2}{M^2} \le \frac{\pi}{M} + \frac{\pi^2}{M^2}$$

with probability at least $8/\pi^2 = 0.81\ldots$. *Hence,*

$$\sum_{j:\, |\bar{a}_f(j)-a_f|\le(2\pi/M)\sqrt{a_f(1-a_f)}+\pi^2/M^2} p_f(j) \ge \frac{8}{\pi^2} \qquad \forall f \in \mathbb{B}_N,$$

and, therefore,

$$e^{\textit{wor-pro}}\left(M, \frac{8}{\pi^2}\right) \le \frac{\pi}{M} + \frac{\pi^2}{M^2}.$$

Using ideas from the proof of Theorem 2.3.1 from [4] we present the following theorem and the subsequent corollaries.

Theorem 2.3.2. *For any Boolean function* $f \in \mathbb{B}_N$, *denote*

$$\sigma_a = \sigma_{a_f} = \frac{M}{\pi}\arcsin\sqrt{a} \in \left[0, \tfrac{1}{2}M\right].$$

1. *The* **QS** *algorithm uses exactly* $M-1$ *quantum queries, and* $\log_2 N + \lceil \log_2 M \rceil$ *qubits.*

2. *For* $j = 0, 1, \ldots, M-1$, *the outcome* j *of the* **QS** *algorithm occurs with probability*

$$p_f(j) = \frac{\sin^2(\pi(j-\sigma_{a_f}))}{2M^2\sin^2\left(\pi(j-\sigma_{a_f})/M\right)}\left(1 + \frac{\sin^2\left(\pi(j-\sigma_{a_f})/M\right)}{\sin^2\left(\pi(j+\sigma_{a_f})/M\right)}\right). \tag{2.6}$$

   (If $\sin\left(\pi(j\pm\sigma_{a_f})/M\right) = 0$ *we need to apply the limiting value of the formula above.) For* $j = M, M+1, \ldots, 2^{\lceil \log_2 M\rceil}-1$, *the outcome* j *occurs with probability* 0.

3. *If* σ_{a_f} *is an integer, then the* **QS** *algorithm outputs the exact value of* a_f *with probability* 1. *This holds iff* $a_f = \sin^2(k\pi/M)$ *for some integer* $k \in [0, \frac{1}{2}M]$. *In particular, this holds for* $a_f = 0$, *for* $a_f = 1$ *and even* M, *and for* $a_f = \frac{1}{2}$ *and* M *divisible by* 4.

4. *Let* $\overline{x} = \pi(\lceil\sigma_a\rceil - \sigma_a)/M$ *and* $\underline{x} = \pi(\sigma_a - \lfloor\sigma_a\rfloor)/M$. *If* σ_{a_f} *is not an integer, then the* **QS** *algorithm outputs the same value* $\bar{a} = \bar{a}_f(\lceil\sigma_a\rceil) = \bar{a}_f(M - \lceil\sigma_a\rceil)$ *for the outcomes* $\lceil\sigma_a\rceil$ *and* $M - \lceil\sigma_a\rceil$ *such that*

$$|\bar{a}-a| = \left|\sin(\overline{x})\left(2\sqrt{a(1-a)}\cos(\overline{x}) + (1-2a)\sin(\overline{x})\right)\right| \le \frac{\pi}{M}\left(\lceil\sigma_a\rceil - \sigma_a\right) \tag{2.7}$$

with probability

$$\frac{\sin^2(\pi(\lceil\sigma_a\rceil-\sigma_a))}{M^2\sin^2\left(\pi(\lceil\sigma_a\rceil-\sigma_a)/M\right)}\left(1+(1-\delta_{\lceil\sigma_a\rceil,M/2})\frac{\sin^2\left(\pi(\lceil\sigma_a\rceil-\sigma_a)/M\right)}{\sin^2\left(\pi(\lceil\sigma_a\rceil+\sigma_a)/M\right)}\right)$$
$$\geq\frac{\sin^2\left(\pi(\lceil\sigma_a\rceil-\sigma_a)\right)}{\pi^2(\lceil\sigma_a\rceil-\sigma_a)^2}=1-\frac{\pi^2}{3}(\lceil\sigma_a\rceil-\sigma_a)^2+O\big((\lceil\sigma_a\rceil-\sigma_a)^4\big),\quad(2.8)$$

and outputs the same value $\bar{a}=\bar{a}_f(\lfloor\sigma_a\rfloor)=\bar{a}_f((1-\delta_{\lfloor\sigma_a\rfloor,0})M-\lfloor\sigma_a\rfloor)$ *for the outcomes* $\lfloor\sigma_a\rfloor$ *and* $(1-\delta_{\lfloor\sigma_a\rfloor,0})M-\lfloor\sigma_a\rfloor$ *such that*

$$|\bar{a}-a|=\left|\sin(\underline{x})\left(2\sqrt{a(1-a)}\cos(\underline{x})+(1-2a)\sin(\underline{x})\right)\right|\leq\frac{\pi}{M}(\sigma_a-\lfloor\sigma_a\rfloor)\quad(2.9)$$

with probability

$$\frac{\sin^2\left(\pi(\sigma_a-\lfloor\sigma_a\rfloor)\right)}{M^2\sin^2\left(\pi(\sigma_a-\lfloor\sigma_a\rfloor)/M\right)}\left(1+(1-\delta_{\lfloor\sigma_a\rfloor,0})\frac{\sin^2\left(\pi(\sigma_a-\lfloor\sigma_a\rfloor)/M\right)}{\sin^2\left(\pi(\sigma_a+\lfloor\sigma_a\rfloor)/M\right)}\right)$$
$$\geq\frac{\sin^2\left(\pi(\sigma_a-\lfloor\sigma_a\rfloor)\right)}{\pi^2(\sigma_a-\lfloor\sigma_a\rfloor)^2}=1-\frac{\pi^2}{3}(\sigma_a-\lfloor\sigma_a\rfloor)^2+O\big((\sigma_a-\lfloor\sigma_a\rfloor)^4\big).\quad(2.10)$$

Proof. As before, let $\theta_a=\arcsin\sqrt{a}$ and

$$|S_M(\omega)\rangle=\frac{1}{\sqrt{M}}\sum_{k=0}^{M-1}e^{2\pi i\omega k}|k\rangle,\quad i=\sqrt{-1},$$

for arbitrary $\omega\in\mathbb{R}$. Note that

$$F_{M,m}|j\rangle=\begin{cases}|S_M(j/M)\rangle & \text{for } j=0,1,\ldots,M-1,\\ |j\rangle & \text{for } j=M,M+1,\ldots,2^m-1.\end{cases}$$

The steps 1–3 of the **QS** algorithm are equivalent to the application of the operator $(F_{M,m}^{-1}\otimes I)\,\Lambda_m(Q_f)\,F_{M,m}\otimes W_N$ to the state $|0\rangle|0\rangle\in\mathcal{H}_m\otimes\mathcal{H}_n$. Now $|\eta_1\rangle$ can be written as $M^{-1/2}\sum_{j=0}^{M-1}|j\rangle|\psi\rangle$, where $|\psi\rangle=W_N|0\rangle$ is given by (2.4). Hence

$$|\eta_1\rangle=\frac{-i}{\sqrt{2M}}\sum_{j=0}^{M-1}|j\rangle\left(e^{i\theta_a}|\psi_+\rangle-e^{-i\theta_a}|\psi_-\rangle\right).$$

Applying $\Lambda_m(Q_f)$ in Step 2 and remembering that $Q_f^j|\psi_\pm\rangle = \lambda_\pm^j|\psi_\pm\rangle$, we obtain

$$\begin{aligned}|\eta_2\rangle = \Lambda_m(Q_f)\,|\eta_1\rangle = \frac{-i}{\sqrt{2M}}\sum_{j=0}^{M-1}|j\rangle\left(e^{2ij\theta_a}e^{i\theta_a}|\psi_+\rangle - e^{-2ij\theta_a}e^{-i\theta_a}|\psi_-\rangle\right)\\ = \frac{-i}{\sqrt{2}}\left(e^{i\theta_a}|S_M(\sigma_a/M)\rangle|\psi_+\rangle - e^{-i\theta_a}|S_M(-\sigma_a/M)\rangle|\psi_-\rangle\right).\end{aligned}$$

Since $j = 0, 1, \ldots, M-1$, the largest power of Q_f is $M-1$. Hence, we use exactly $M-1$ quantum queries to compute $|\eta_2\rangle$. The remaining steps of the **QS** algorithm do not use quantum queries. This means that the total number of quantum queries used by the QS algorithm is $M-1$, and obviously we are using $n+m$ qubits. This proves the first part of Theorem 2.3.2.

Step 3 yields the state

$$\begin{aligned}|\eta_3\rangle &= (F_{M,m}^{-1}\otimes I)|\eta_2\rangle\\ &= \frac{-i}{\sqrt{2}}\left(e^{i\theta_a}F_{M,m}^{-1}|S_M(\sigma_a/M)\rangle|\psi_+\rangle - e^{-i\theta_a}F_{M,m}^{-1}|S_M(-\sigma_a/M)\rangle|\psi_-\rangle\right).\end{aligned}$$

We are ready to analyze the probability of the outcome j of the **QS** algorithm. Observe that

$$\begin{aligned}|\alpha_\pm\rangle := (|j\rangle\langle j|\otimes I)\ F_{M,m}^{-1}|S_M(\pm\sigma_a/M)\rangle|\psi_\pm\rangle = \langle j|F_{M,m}^{-1}|S_M(\pm\sigma_a/M)\rangle\,|j\rangle|\psi_\pm\rangle\\ = \begin{cases}\langle S_M(j/M)|S_M(\pm\sigma_a/M)\rangle\,|j\rangle|\psi_\pm\rangle & \text{for } j = 0, 1, \ldots, M-1,\\ 0 & \text{for } j = M, M+1, \ldots, 2^m-1,\end{cases}\end{aligned}$$

and therefore

$$\langle\alpha_\pm|\alpha_\pm\rangle = \begin{cases}|\langle S_M(j/M)|S_M(\pm\sigma_a/M)\rangle|^2\,\langle\psi_\pm|\psi_\pm\rangle & \text{for } j = 0, 1, \ldots, M-1.\\ 0 & \text{for } j = M, \ldots, 2^m-1.\end{cases}$$

Observe that for $a \in (0,1)$, we have $\langle\psi_\pm|\psi_\pm\rangle = 1$, whereas for $a \in \{0,1\}$, we have $\langle\psi_+|\psi_+\rangle = 2$ and $\langle\psi_-|\psi_-\rangle = 0$.

For $\omega_1, \omega_2 \in \mathbb{R}$ we have

$$|\langle S_M(\omega_1)|S_M(\omega_2)\rangle|^2 = \left|\left(\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1} e^{-2\pi i\omega_1 j}\langle j|\right)\left(\frac{1}{\sqrt{M}}\sum_{j=0}^{M-1} e^{2\pi i\omega_2 j}|j\rangle\right)\right|^2$$
$$= \frac{1}{M^2}\left|\sum_{j=0}^{M-1} e^{-2\pi i(\omega_1-\omega_2)j}\right|^2.$$

If $\omega_1 - \omega_2$ is an integer, then the last sum is clearly M, and the whole expression is 1. If $\omega_1 - \omega_2$ is not an integer, then

$$\frac{1}{M}\sum_{j=0}^{M-1} e^{-2\pi i(\omega_1-\omega_2)j} = \frac{e^{-2\pi iM(\omega_1-\omega_2)}-1}{M(e^{-2\pi i(\omega_1-\omega_2)}-1)},$$

which holds for all $\omega_1, \omega_2 \in \mathbb{R}$ if we take 0/0 as 1. Therefore

$$\left|\frac{1}{M}\sum_{j=0}^{M-1} e^{-2\pi i(\omega_1-\omega_2)j}\right|^2 = \frac{1-\cos(2\pi M(\omega_1-\omega_2))}{M^2(1-\cos(2\pi(\omega_1-\omega_2)))} = \frac{\sin^2(\pi M(\omega_1-\omega_2))}{M^2(\sin^2(\pi(\omega_1-\omega_2)))}.$$

Hence

$$|\langle S_M(\omega_1)|S_M(\omega_2)\rangle|^2 = \frac{\sin^2(M\pi(\omega_1-\omega_2))}{M^2\sin^2(\pi(\omega_1-\omega_2))}, \tag{2.11}$$

which holds for all $\omega_1, \omega_2 \in \mathbb{R}$ if we take 0/0 as 1. Applying this we conclude that

$$\langle\alpha_\pm|\alpha_\pm\rangle = \begin{cases} \dfrac{\sin^2(\pi(j\mp\sigma_a))}{M^2\sin^2(\pi(j\mp\sigma_a)/M)}\langle\psi_\pm|\psi_\pm\rangle & \text{for } j=0,1,\ldots,M-1,\\ 0 & \text{for } j=M,M+1,\ldots,2^m-1.\end{cases}$$

The outcome j occurs after the measurement and the state $|\eta_3\rangle$ collapses to the state

$(\langle\eta_3|M_j^\dagger M_j|\eta_3\rangle)^{-1}M_j|\eta_3\rangle$, where $M_j = |j\rangle\langle j|\otimes I$. For $j=0,1,\ldots,M-1$, we have

$$M_j|\eta_3\rangle = |j\rangle\Bigg(\frac{-i}{\sqrt{2}}\big(e^{i\theta_a}\langle S_M(j/M)|S_M(\sigma_a/M)\rangle\,|\psi_+\rangle$$
$$- e^{-i\theta_a}\langle S_M(j/M)|S_M(-\sigma_a/M)\rangle\,|\psi_-\rangle\big)\Bigg),$$

whereas $M_j|\eta_3\rangle = 0$ for $j = M, M+1,\ldots,2^m-1$. Since $|\psi_+\rangle$ and $|\psi_-\rangle$ are orthogonal we have

$$\langle\eta_3|M_j^\dagger M_j|\eta_3\rangle = \tfrac{1}{2}\left(\langle\alpha_+|\alpha_+\rangle + \langle\alpha_-|\alpha_-\rangle\right).$$

We now claim that for any $j \in \{0, 1, \dots, M-1\}$, the outcome j occurs with probability

$$p_f(j) = \frac{1}{2}\left(\frac{\sin^2(\pi(j-\sigma_a))}{M^2\sin^2(\pi(j-\sigma_a)/M)} + \frac{\sin^2(\pi(j+\sigma_a))}{M^2\sin^2(\pi(j+\sigma_a)/M)}\right). \tag{2.12}$$

Indeed, for $a \in (0,1)$, we have $\langle\psi_\pm|\psi_\pm\rangle = 1$ and (2.12) follows from the form of $\langle\alpha_\pm|\alpha_\pm\rangle$. For $a \in \{0,1\}$, we have $\langle\psi_+|\psi_+\rangle = 2$ and $\langle\psi_-|\psi_-\rangle = 0$. Since the two terms in (2.12) are now the same, the formula for $\langle\alpha_+|\alpha_+\rangle$ again yields (2.12).

Since $\sin^2(\pi(j-\sigma_a)) = \sin^2(\pi(j+\sigma_a))$, the last formula is equivalent to (2.6). Obviously for $j = M, M+1, \dots, 2^m - 1$, the probability of the outcome j is zero. This proves the second part of Theorem 2.3.2.

Assume now that $\sigma_a \in \mathbb{Z}$. If $\sigma_a = 0$ or $\sigma_a = \frac{1}{2}M$ (if M is even), then the probability $p_f(\sigma_a)$ of the outcome σ_a is 1. For $\sigma_a = 0$ we have $a = 0$ and the output is $\bar{a}_f(0) = 0$. For $\sigma_a = \frac{1}{2}M$ we have $a = 1$ and the output is $\bar{a}_f(\frac{1}{2}M) = 1$. Hence, in both cases the **QS** algorithm outputs the exact value with probability 1.

If $\sigma_a \in \mathbb{Z}$ and $\sigma_a \notin \{0, \frac{1}{2}M\}$, then the probability of the distinct outcomes σ_a and $M - \sigma_a$ is $\frac{1}{2}$. These two values of the outcomes yield the same output

$$\sin^2(\pi\sigma_a/M) = \sin^2(\pi(M-\sigma_a)/M) = a.$$

Hence, the **QS** algorithm outputs the exact value with probability 1. This proves the third part of Theorem 2.3.2.

We now turn to the case when $\sigma_a \notin \mathbb{Z}$. It is easy to check that the third part of Theorem 2.3.2 holds for $M = 1$. Assume then that $M \geq 2$, which implies that $\lceil \frac{1}{2}M \rceil \leq M-1$. Since σ_a is not an integer, we have $\lceil\sigma_a\rceil \geq 1$, $\lceil\sigma_a\rceil \leq \lceil\frac{1}{2}M\rceil \leq M-1$ and $M - \lceil\sigma_a\rceil \leq M-1$. This means that both $\lceil\sigma_a\rceil$ and $M-\lceil\sigma_a\rceil$ may be the outcomes of the **QS** algorithm. Obviously, these two outcomes are different iff $\lceil\sigma_a\rceil \neq \frac{1}{2}M$. Similarly, both $\lfloor\sigma_a\rfloor$ and $(1-\delta_{\lfloor\sigma_a\rfloor,0})M - \lfloor\sigma_a\rfloor$ may be also the outcomes. They are different iff $\lfloor\sigma_a\rfloor \neq 0$.

We show that the outputs for the outcomes $\lceil\sigma_a\rceil$ and $\lfloor\sigma_a\rfloor$ satisfy (2.7) and (2.9) with probabilities (2.8) and (2.10), respectively. We focus on the output for the outcome $\lceil \sigma_a \rceil$ and its probability. The proof for the outcome $\lfloor \sigma_a \rfloor$ is similar.

We estimate the error of the **QS** algorithm for the output $\bar{a} = \sin^2(\pi \lceil \sigma_a \rceil / M)$. Recall that $\overline{x} = \pi(\lceil \sigma_a \rceil - \sigma_a)/M$. We have

$$\begin{aligned} |\bar{a} - a| = |\sin^2(\pi \lceil \sigma_a \rceil / M) - \sin^2(\pi \sigma_a / M)| &= |\sin(\overline{x}) \, \sin(\overline{x} + 2\pi\sigma_a/M)| \\ = \bigl| \sin(\overline{x}) \, \bigl(\sin(2\pi\sigma_a/M) \cos(\overline{x}) &+ \cos(2\pi\sigma_a/M) \sin(\overline{x}) \bigr) \bigr| \\ &\leq \pi(\lceil \sigma_a \rceil - \sigma_a)/M. \end{aligned}$$

Since $\sin(2\pi\sigma_a/M) = 2\sqrt{a(1-a)}$ and $\cos(2\pi\sigma_a/M) = 1 - 2a$, this proves the estimate of the error of the **QS** algorithm in the fourth part of Theorem 2.3.2.

We find the probability of the output $\bar{a}$. Since $t \mapsto \sin^2(\pi t/M)$ is injective for $t \in [0, \frac{1}{2}M]$, the output $\bar{a}$ occurs only for the outcomes $\lceil \sigma_a \rceil$ and $M - \lceil \sigma_a \rceil$. If $\lceil \sigma_a \rceil = \frac{1}{2}M$, then these two outcomes are the same and $\bar{a}$ occurs with probability $p_f(\frac{1}{2}M)$. Due to (2.12), we have

$$p_f\left(\tfrac{1}{2}M\right) = \frac{\sin^2\left(\pi(\frac{1}{2}M - \sigma_{a_f})\right)}{M^2 \sin^2\left(\pi(\frac{1}{2}M - \sigma_{a_f})/M\right)},$$

which agrees with the claim in Theorem 2.3.2.

If $\lceil \sigma_a \rceil \neq \frac{1}{2}M$, then $\lceil \sigma_a \rceil \neq M - \lceil \sigma_a \rceil$ and $\bar{a}$ occurs for exactly two distinct outcomes. The probability of $\bar{a}$ is now equal to the sum of the probabilities $p_f(\lceil \sigma_a \rceil) + p_f(M - \lceil \sigma_a \rceil)$, where p_f is given by (2.12). Since both terms are equal, the probability of $\bar{a}$ is $2p_f(\lceil \sigma_a \rceil)$ which also agrees with the claim in Theorem 2.3.2. Since $\sin\left(\pi(\lceil \sigma_a \rceil - \sigma_a)/M\right) \leq \pi(\lceil \sigma_a \rceil - \sigma_a)/M$ we have

$$\frac{\sin^2(\pi(\lceil \sigma_a \rceil - \sigma_a))}{M^2 \sin^2\left(\pi(\lceil \sigma_a \rceil - \sigma_a)/M\right)} \geq \frac{\sin^2(\pi \lceil \sigma_a \rceil - \sigma_a)}{\pi^2(\lceil \sigma_a \rceil - \sigma_a)^2}.$$

We finish proving (2.8) by using the standard expansion of the sine. This completes the proof. □

Based on Theorem 2.3.2, we present simplified estimates of the error of the **QS** algorithm and of the corresponding probability.

Corollary 2.3.3. *The* **QS** *algorithm outputs $\bar{a}$ such that*

$$|\bar{a} - a| \leq \frac{\pi}{M} \max\{\lceil \sigma_a \rceil - \sigma_a, \sigma_a - \lfloor \sigma_a \rfloor\} \tag{2.13}$$

with probability at least

$$\frac{\sin^2(\pi(\lceil \sigma_a \rceil - \sigma_a))}{M^2 \sin^2 (\pi(\lceil \sigma_a \rceil - \sigma_a)/M)} + \frac{\sin^2(\pi(\sigma_a - \lfloor \sigma_a \rfloor))}{M^2 \sin^2 (\pi(\sigma_a - \lfloor \sigma_a \rfloor)/M)} \geq \frac{8}{\pi^2}. \tag{2.14}$$

Proof. It is enough to prove Corollary 2.3.3 if σ_a is not an integer. The estimate of the error of the **QS** algorithm by the maximum of the estimates (2.7) and (2.9) holds with probability that is the sum of the probabilities (2.8) and (2.10). Moreover, $\lceil \sigma_a \rceil - \sigma_a = 1 - (\sigma_a - \lfloor \sigma_a \rfloor)$. Observe that

$$g(\Delta) = \frac{\sin^2(\pi\Delta)}{\pi^2\Delta^2} + \frac{\sin^2(\pi(1-\Delta))}{\pi^2(1-\Delta)^2}$$

is a lower bound of the left hand side of (2.14) with $\Delta = \lceil \sigma_a \rceil - \sigma_a$. Since the function g attains the minimum $8/\pi^2$ on the interval $[0, 1]$ for $\Delta = \frac{1}{2}$, see also [4], this completes the proof. □

Corollary 2.3.3 guarantees that the estimate (2.13) holds with high probability. Unfortunately this estimate does not preserve the continuity of the estimates (2.7) and (2.9) with respect to $\lceil \sigma_a \rceil - \sigma_a$ and $\sigma_a - \lfloor \sigma_a \rfloor$. The continuity of the estimates is present in the next corollary, at the expense of the probability of the outcome. This corollary will also play an essential role in the study of the average-probabilistic error of the **QS** algorithm.

Corollary 2.3.4. *The* **QS** *algorithm outputs $\bar{a}$ such that*

$$|\bar{a} - a| \leq \frac{\pi}{M} \min\{\lceil \sigma_a \rceil - \sigma_a, \sigma_a - \lfloor \sigma_a \rfloor\} \tag{2.15}$$

with probability at least

$$\max\left\{ \frac{\sin^2(\pi(\lceil \sigma_a \rceil - \sigma_a))}{M^2 \sin^2(\frac{\pi}{M}(\lceil \sigma_a \rceil - \sigma_a))}, \frac{\sin^2(\pi(\sigma_a - \lfloor \sigma_a \rfloor))}{M^2 \sin^2(\frac{\pi}{M}(\sigma_a - \lfloor \sigma_a \rfloor))} \right\} \geq \frac{4}{\pi^2}. \tag{2.16}$$

Proof. We may again assume that σ_a is not an integer. Let us define

$$w(\Delta) = \frac{\sin^2(\pi\Delta)}{M^2 \sin^2(\pi\Delta/M)} \qquad \text{for} \quad \Delta \in [0,1].$$

Then $w(\lceil\sigma_a\rceil - \sigma_a)$ is the probability that (2.7) holds and $w(1 - (\lceil\sigma_a\rceil - \sigma_a))$ is the probability that (2.9) holds. For $\Delta \in [0, \frac{1}{2}]$, note that $w(\cdot)$ is decreasing, and $w(1 - \cdot)$ is increasing. Therefore

$$w(\Delta) \geq w(\tfrac{1}{2}) \geq w(1-\Delta) \qquad \text{for} \quad \Delta \in [0, \tfrac{1}{2}].$$

Suppose that $\lceil\sigma_a\rceil - \sigma_a \leq \sigma_a - \lfloor\sigma_a\rfloor$. Then $\lceil\sigma_a\rceil - \sigma_a \leq \frac{1}{2}$. In this case (2.15) is equivalent to (2.7), and holds with probability at least $w(\lceil\sigma_a\rceil - \sigma_a)$, which corresponds to (2.16). Analogously, if $\lceil\sigma_a\rceil - \sigma_a \geq \sigma_a - \lfloor\sigma_a\rfloor$ then $\lceil\sigma_a\rceil - \sigma_a \geq \frac{1}{2}$. In this case (2.15) is equivalent to (2.9), and holds with probability at least $w(\sigma_a - \lfloor\sigma_a\rfloor)$, which also corresponds to (2.16). Finally, note that

$$\max\{w(\lceil\sigma_a\rceil - \sigma_a), w(\sigma_a - \lfloor\sigma_a\rfloor)\}$$

is minimal for $\lceil\sigma_a\rceil - \sigma_a = \frac{1}{2}$ and is equal to

$$\frac{1}{M^2} \sin^{-2} \frac{\pi}{2M} \geq \frac{4}{\pi^2}.$$

□

Unfortunately, for $\lceil\sigma_a\rceil - \sigma_a$ close to $\frac{1}{2}$ the probability of the estimate (2.15) is too small. However in this case we may use Corollary 2.3.3, which yields an estimate with high probability.

We now turn to global error estimates, that is, estimates independent of a. Theorem 2.3.1 of [4] states, in particular, that $|\bar{a} - a| \leq \pi/M + \pi^2/M^2$ with probability at least $8/\pi^2$. We now improve this estimate by combining the estimates (2.13) and (2.15).

Corollary 2.3.5. *The* **QS** *algorithm outputs* $\bar{a}$ *such that*

$$|\bar{a} - a| \le \frac{3}{4}\,\frac{\pi}{M} \tag{2.17}$$

with probability at least $8/\pi^2$. *That is,*

$$e^{wor\text{-}pro}\left(M, \frac{8}{\pi^2}\right) \le \frac{3}{4}\,\frac{\pi}{M}.$$

Proof. Let us define

$$h(\Delta) = \max\left\{\frac{\sin^2(\pi\Delta)}{\pi^2\Delta^2}, \frac{\sin^2(\pi(1-\Delta))}{\pi^2(1-\Delta)^2}\right\}. \tag{2.18}$$

Clearly, $h(\lceil\sigma_a\rceil - \sigma_a)$ is a lower bound on $\max\{w(\lceil\sigma_a\rceil - \sigma_a), w(1-(\lceil\sigma_a\rceil - \sigma_a))\}$ and therefore $h(\lceil\sigma_a\rceil - \sigma_a)$ is a lower bound of the probability of the output satisfying (2.15). We consider two cases.

Assume first that $\Delta = \lceil\sigma_a\rceil - \sigma_a \in [0, \frac{1}{4}] \cup [\frac{3}{4}, 1]$. It is easy to see that then $h(\Delta) \ge 8/\pi^2$ and the estimate (2.15) yields

$$|\bar{a} - a| \le \frac{\pi}{M}\min\{\lceil\sigma_a\rceil - \sigma_a, \sigma_a - \lfloor\sigma_a\rfloor\} \le \frac{1}{4}\,\frac{\pi}{M}$$

with probability at least $8/\pi^2$.

Assume now that $\lceil\sigma_a\rceil - \sigma_a \in (\frac{1}{4}, \frac{3}{4})$. Then we can use the estimate (2.13), which holds unconditionally with probability at least $8/\pi^2$. In this case, we have

$$|\bar{a} - a| \le \frac{\pi}{M}\max\{\lceil\sigma_a\rceil - \sigma_a, \sigma_a - \lfloor\sigma_a\rfloor\} \le \frac{3}{4}\,\frac{\pi}{M}.$$

These two estimates combined together yield (2.17). □

An obvious consequence of Corollary 2.3.5 is that for M large enough we can compute the value of a exactly by rounding the output.

Corollary 2.3.6. *Assume that*

$$M > \frac{3\pi}{2}N.$$

Then the rounding of the **QS** *algorithm output to the nearest number of the form* k/N *yields the exact value of the sum* a *with probability at least* $8/\pi^2$.

The proof of Corollary 2.3.5 may suggest that the constant $\frac{3}{4}$ in (2.17) can be decreased. Furthermore one may want to decrease the constant $\frac{3}{4}$ at the expense of decreasing the probability $8/\pi^2$. These points are addressed in the next corollary. We shall see that the constant $\frac{3}{4}$ may be lowered only by decreasing the probability.

Corollary 2.3.7. *Define*

$$C(p) = \inf\left\{ C: \ |\bar{a}_f - a_f| \le C\frac{\pi}{M} \quad \forall f \in \mathbb{B}_N \text{ with probability at least } p \right\}$$

and

$$v(\Delta) = \frac{\sin^2(\pi\Delta)}{\pi^2\Delta^2} \qquad \text{for } \Delta \in [\tfrac{1}{4}, \tfrac{1}{2}].$$

Then

$$C(p) \le \begin{cases} \frac{1}{2} & \text{for } p \in [0, 4/\pi^2), \\ 1 - v^{-1}(p) & \text{for } p \in [4/\pi^2, 8/\pi^2], \\ M/\pi & \text{for } p \in (8/\pi^2, 1]. \end{cases} \tag{2.19}$$

Moreover, $1 - v^{-1}(p) \in [\frac{1}{2}, \frac{3}{4}]$ *and*

$$\left| \frac{\pi^2}{16}p + \frac{1}{4} - \left(1 - v^{-1}(p)\right) \right| \le 0.0085 \quad \text{for } p \in [4/\pi^2, 8/\pi^2]. \tag{2.20}$$

Proof. For $p \in [0, 4/\pi^2)$, Corollary 2.3.7 is a consequence of Corollary 2.3.4. For $p \in (8/\pi^2, 1]$, Corollary 2.3.7 trivially holds since $|\bar{a} - a| \le 1 = (M/\pi)\pi/M$. For the remaining p's we use a proof technique similar to that of Corollary 2.3.5.

Let $p \in [4/\pi^2, 8/\pi^2]$. It is easy to check that v is decreasing and, therefore, $v^{-1}(p)$ is well defined and $v^{-1}(p) \in [\frac{1}{4}, \frac{1}{2}]$. We have to show that the estimate

$$|\bar{a} - a| \le \left(1 - v^{-1}(p)\right)\frac{\pi}{M} \tag{2.21}$$

holds with probability at least p. We consider two cases.

Assume first that $\Delta = \lceil \sigma_a \rceil - \sigma_a \in [0, v^{-1}(p)] \cup [1 - v^{-1}(p), 1]$. Observe that the function h defined in (2.18) can be rewritten as

$$h(\Delta) = \max\{v(\Delta), v(1 - \Delta)\}.$$

It is easy to see that in this case, $h(\Delta) \geq p$, and the estimate (2.15) yields

$$|\bar{a} - a| \leq \frac{\pi}{M} \min\{\Delta, 1-\Delta\} \leq v^{-1}(p)\, \frac{\pi}{M} \leq \big(1 - v^{-1}(p)\big) \frac{\pi}{M}$$

with probability at least p.

Assume now that $\Delta = \lceil \sigma_a \rceil - \sigma_a \in \big(v^{-1}(p), 1 - v^{-1}(p)\big)$. Then we can use the estimate (2.13), which holds unconditionally with probability at least $8/\pi^2 > p$. In this case, we have

$$|\bar{a} - a| \leq \frac{\pi}{M} \max\{\Delta, 1-\Delta\} \leq \big(1 - v^{-1}(p)\big) \frac{\pi}{M}.$$

This proves (2.21).

We found the estimate (2.20) by numerical computations. □

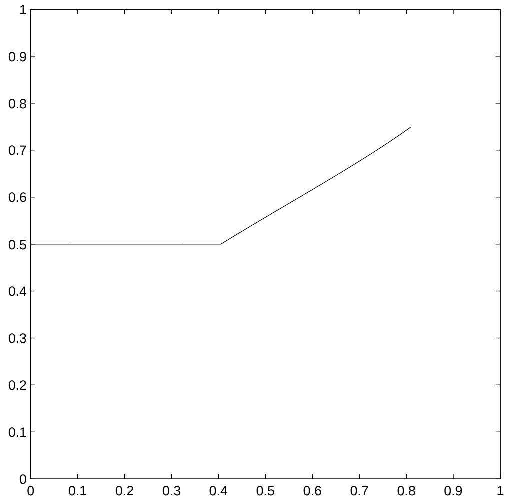

Figure 2.1: The estimate (2.19) of $C(p)$ for $p \in [0, 8/\pi^2]$

From Figure 2.1 we see that the estimate (2.19) is almost linear on the interval $[4/\pi^2, 8/\pi^2]$, which explains why the right hand side of the estimate (2.20) is small.

We now find a sharp bound on the worst-probabilistic error of the **QS** algorithm. We show that for large M and N/M the bound obtained in Corollary 2.3.7 is optimal for $p \in (1/2, 8/\pi^2]$.

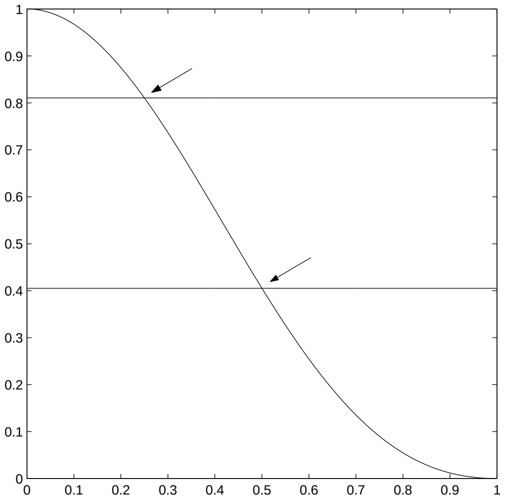

Figure 2.2: The function v on $[0, 1]$. The two horizontal lines show $4/\pi^2$ and $8/\pi^2$ levels. The part of the graph between the arrows shows that v is almost linear.

Theorem 2.3.8. *For large M and N/M, the worst-probabilistic error of the* **QS** *algorithm is given by*

$$e^{\text{wor-pro}}(M,p) \;=\; \big(1 - v^{-1}(p)\big)\frac{\pi}{M}\left(1 + O(M^{-1}) + O(MN^{-1})\right) \qquad \text{for } p \in (\tfrac{1}{2}, \tfrac{8}{\pi^2}].$$

Here, v is as in Corollary 2.3.7, and $1 - v^{-1}(p) \approx \frac{1}{16}\pi^2 p + \frac{1}{4}$ by (2.20).

Proof. From Corollary 2.3.7, it is enough to show a lower bound on the error. Define

$$s_1 = \sin^2\left(\frac{\pi\left\lceil \frac{1}{4}M\right\rceil}{M}\right) \quad \text{and} \quad s_2 = \sin^2\left(\frac{\pi\big(1 + \left\lceil\frac{1}{4}M\right\rceil\big)}{M}\right).$$

For large M, we have

$$s_i = \tfrac{1}{2} + O(M^{-1})$$

and

$$s_2 - s_1 = \sin\left(\frac{\pi}{M}\right)\sin\left(\frac{\pi(1 + 2\left\lceil\frac{1}{4}M\right\rceil)}{M}\right) = \frac{(1 + O(M^{-1}))\pi}{M}.$$

There exist two Boolean functions f_1 and f_2 with sums $a_1 = a_{f_1}$ and $a_2 = a_{f_2}$ such that

$$|a_i - s_i| \le N^{-1} \quad \text{for} \quad i = 1, 2.$$

Since $\sigma_{s_i} = \left\lceil \frac{1}{4}M \right\rceil + (i-1)$ and the derivative of σ_a for $a = \frac{1}{2}$ is M/π, we have

$$\sigma_{a_1} = \left\lceil \tfrac{1}{4}M \right\rceil + O(MN^{-1}) \quad \text{and} \quad \sigma_{a_2} = \left\lceil \tfrac{1}{4}M \right\rceil + 1 + O(MN^{-1}).$$

Obviously, $a_i = k_i/N$ for some integers k_i with $k_1 < k_2$. Consider $\sigma_{x/N}$ for $x \in \{k_1, k_1 + 1, \ldots, k_2\}$. Then $\sigma_{x/N}$ varies from σ_{a_1} for $x = k_1$ to σ_{a_2} for $x = k_2$. Since $v^{-1}(p) \in [\frac{1}{4}, \frac{1}{2})$, for a positive and small η (we finally let η go to zero), we can choose $x = x_\eta$ such that for $a^* = x_\eta/N$ we have

$$\sigma_{a^*} := \left\lceil \tfrac{1}{4}M \right\rceil + v^{-1}(p) + \eta + O(MN^{-1}).$$

For large N/M, we then have

$$\lfloor \sigma_{a^*} \rfloor = \left\lceil \tfrac{1}{4}M \right\rceil, \quad \lceil \sigma_{a^*} \rceil = \left\lceil \tfrac{1}{4}M \right\rceil + 1$$

and

$$\sigma_{a^*} - \lfloor \sigma_{a^*} \rfloor = v^{-1}(p) + \eta + O(M/N), \;\; \lceil \sigma_{a^*} \rceil - \sigma_{a^*} = 1 - v^{-1}(p) - \eta + O(M/N).$$

Let $\bar{a}_1^*$ denote the output for the outcome $\lceil \sigma_{a^*} \rceil$, and $\bar{a}_2^*$ for $\lfloor \sigma_{a^*} \rfloor$.

Due to (2.7) and (2.9), of Theorem 2.3.2 we have

$$\begin{aligned} |a^* - \bar{a}_1^*| &= \frac{\pi}{M}\left(1 - v^{-1}(p) - \eta\right)\left(1 + O(M^{-1} + MN^{-1})\right) \\ |a^* - \bar{a}_2^*| &= \frac{\pi}{M}\left(v^{-1}(p) + \eta\right)\left(1 + O(M^{-1} + MN^{-1})\right). \end{aligned}$$

Let us write $1 + o(1)$ for $1 + O(\eta^2 + M^{-1} + MN^{-1})$. The probability of $\bar{a}_2^*$ is given by (2.10) and is now equal to

$$\frac{\sin^2(\pi(v^{-1}(p) + \eta))}{(\pi(v^{-1}(p) + \eta))^2}\left(1 + o(1)\right) = \frac{\sin^2(\pi v^{-1}(p)) + \pi\eta \sin(2\pi v^{-1}(p))}{\pi^2 v^{-1}(p)^2 (1 + 2\eta/v^{-1}(p))}\left(1 + o(1)\right).$$

Since $p = v(v^{-1}(p)) = \sin^2(\pi v^{-1}(p))/(\pi v^{-1}(p))^2$, the probability of $\bar{a}_2^*$ is

$$p\left(1 - 2\eta\left(\frac{1}{v^{-1}(p)} - \pi\cot(\pi v^{-1}(p))\right)\right)(1 + o(1)).$$

Since $\cot(t) < 1/t$ for $t \in [\frac{1}{4}\pi, \frac{1}{2}\pi]$, we see that the probability of $\bar{a}_2^*$ is slightly less than p for small η.

We are ready to find a lower bound on the worst-probabilistic error

$$e^{\text{wor-pro}}(M,p) = \max_{f\in\mathbb{B}_N} \min_{A\colon \mu(A,f)\geq p} \max_{j\in A} |a_f - \bar{a}_f(j)|$$

of the **QS** algorithm. Take the function f that corresponds to a^*. We claim that the error is minimized if $A = \{\lfloor\sigma_{a^*}\rfloor, \lceil\sigma_{a^*}\rceil\}$. Indeed, $\lfloor\sigma_{a^*}\rfloor$ must belong to A since otherwise $\mu(A,f) \leq 1-p_f(\lfloor\sigma_{a^*}\rfloor) = 1-p+o(1) < p$ for $p > \frac{1}{2}$. The probability of $\lfloor\sigma_{a^*}\rfloor$ is slightly less than p, and so the set A must also contain some other outcome j. If $j = \lceil\sigma_{a^*}\rceil$, then the error bound is roughly $(1 - v^{-1}(p) - \eta)\pi/M$, and the sum of the probabilities of the outputs for the outcome $\lfloor\sigma_{a^*}\rfloor$ and $\lceil\sigma_{a^*}\rceil$ is always at least $8/\pi^2 \geq p$. On the other hand, if $\lceil\sigma_{a^*}\rceil$ does not belong to the set A, then any other outcome j yields the output $\sin^2(\pi j/M)$. Since $\sin^2(\pi(j+1)/M) - \sin^2(\pi j/M) = \sin(\pi/M)\sin(\pi(2j+1)/M)$, the distribution of the outcomes around $\frac{1}{2}$ is a mesh with step size roughly π/M. Hence, if $j \neq \lceil\sigma_{a^*}\rceil$, the error is at least roughly $(1 + v^{-1}(p))\pi/M > \pi(1 - v^{-1}(p))/M$. Thus the choice $j = \lceil\sigma_{a^*}\rceil$ minimizes the error and for η tending to zero, the error is roughly $(1 - v^{-1}(p))\pi/M$. This completes the proof. □

From these results, it is obvious how to guarantee that the error of the **QS** algorithm is at most ε with probability at least p. Since $|\bar{a} - a| \leq \big(1 - v^{-1}(p)\big)\pi/M$ holds with probability p, it is enough to take $M \geq \big(1 - v^{-1}(p)\big)\pi/\varepsilon$. By Theorem 2.3.8, this bound is sharp for small ε and large εN. We have

Corollary 2.3.9. *For $p \in (\frac{1}{2}, \frac{8}{\pi^2}]$, the algorithm* **QS**$(f, \lceil(1 - v^{-1}(p))\pi/\varepsilon\rceil)$ *computes $\bar{a}$ with the error ε and probability at least p with $\big\lceil\big(1 - v^{-1}(p)\big)\pi/\varepsilon\big\rceil - 1$ quantum queries. For small ε and large εN, the estimate of the number of quantum queries is sharp.*

2.3.2 Average-probabilistic error

In this section, we study the average performance of the **QS** algorithm with respect to some measure on the set $\mathbb{B}_N$ of all Boolean functions defined on the set $\{0,\dots,N-1\}$.

We consider two such measures. The first measure $\mathbf{p}_1$ is uniformly distributed on the set $\mathbb{B}_N$, i.e.,

$$\mathbf{p}_1(f) = 2^{-N} \qquad \forall f \in \mathbb{B}_N.$$

The second measure $\mathbf{p}_2$ is uniformly distributed on the set of results, i.e.,

$$\mathbf{p}_2(f) = \frac{1}{\binom{N}{k}(N+1)} \quad \text{if} \quad a_f = \frac{k}{N}.$$

We want to estimate average-probabilistic errors

$$e^{\text{avg-pro}}_{\mathbf{p}_i}(M,p) = \sum_{f \in \mathbb{B}_N} \mathbf{p}_i(f) \min_{A:\ \mu(A,f) \ge p} \max_{j \in A} |a_f - a_f(j)| \quad \text{for } i = 1,2.$$

For the measures $\mathbf{p}_i$, the mean of the random variable a_f is clearly $\frac{1}{2}$. However, their first (central) moments are very different. As we shall see, the moment for the measure $\mathbf{p}_1$ is small since it is of order $N^{-1/2}$, whereas the moment for measure $\mathbf{p}_2$ is roughly $\frac{1}{4}$. Since the first moments are the same as the error of the constant algorithm $\bar{a}_f(j) = \frac{1}{2}$, we can achieve small error of order $N^{-1/2}$ for the measure $\mathbf{p}_1$ without any quantum queries, while this property does not hold for the measure $\mathbf{p}_2$.

We first consider the measure $\mathbf{p}_1$. It is natural to ask if the average-probabilistic error of the **QS** algorithm features the dependence of N similar to that of the constant algorithm $\bar{a}_f(j) = \frac{1}{2}$. We shall prove that this is indeed the case iff M is divisible by 4.

We compute the first moment (the error) of the constant algorithm, which is

$$\sum_{k=0}^{N} 2^{-N} \binom{N}{k} \left| \frac{1}{2} - \frac{k}{N} \right|.$$

We do this only for odd N, the case of even N being analogous. We have

$$\begin{aligned}\sum_{k=0}^{N} 2^{-N} \binom{N}{k} \left| \frac{1}{2} - \frac{k}{N} \right| &= 2 \sum_{k=0}^{\lfloor N/2 \rfloor} \binom{N}{k} \left(\frac{1}{2} - \frac{k}{N} \right) = \sum_{k=0}^{\lfloor N/2 \rfloor} \binom{N}{k} - 2 \sum_{k=0}^{\lfloor N/2 \rfloor - 1} \binom{N-1}{k} \\ &= 2^{N-1} - 2 \times \frac{1}{2} \left(2^{N-1} - \binom{N-1}{(N-1)/2} \right) = \binom{N-1}{(N-1)/2}.\end{aligned}$$

Thus

$$e_{\mathbf{p}_1}^{\text{avg-pro}}(0,1) = \sum_{k=0}^{N} 2^{-N}\binom{N}{k}\left|\frac{1}{2}-\frac{k}{N}\right| = \begin{cases} 2^{-N}\dbinom{N-1}{(N-1)/2} & \text{if } N \text{ is odd,} \\ 2^{-(N+1)}\dbinom{N}{N/2} & \text{if } N \text{ is even.} \end{cases} \tag{2.22}$$

Using Stirling's formula

$$k! = \sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{\theta_k/12k} \qquad \text{for some } \theta_k \in [0,1]\ ,$$

we can estimate any of the binomial quantities in (2.22) by

$$\frac{1}{\sqrt{2\pi}}\,\frac{1}{\sqrt{N-1}}\ e^{1/(12(N-1))}$$

proving that

$$\begin{aligned} e_{\mathbf{p}_1}^{\text{avg-pro}}(0,1) = \sum_{k=0}^{N} 2^{-N}\binom{N}{k}\left|\frac{1}{2}-\frac{k}{N}\right| &\le \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{N-1}}\,e^{1/(12(N-1))} \\ &= \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{N}}(1+o(1)). \end{aligned} \tag{2.23}$$

We are ready to analyze the average-probabilistic error of the **QS** algorithm.

Theorem 2.3.10. *Assume that M is divisible by 4 and let $p \in (1/2, 8/\pi^2]$. Then the average-probabilistic error of the* **QS** *algorithms with respect to the measure* $\mathbf{p}_1$ *satisfies*

$$\begin{aligned} e_{\mathbf{p}_1}^{\textit{avg-pro}}(M,p) &\le \min\left\{\frac{3}{4}\,\frac{\pi}{M}, \sqrt{\frac{3}{2\pi}}\sqrt{1+\frac{\pi^2}{4M^2}}\,\frac{1}{\sqrt{N-1}}e^{1/(12(N-1))}\right\} \\ &\le \frac{3}{4}\pi(1+o(1))\min\left\{\frac{1}{M}, \frac{1}{\sqrt{N}}\right\}. \end{aligned}$$

Proof. The estimate $e_{\mathbf{p}_1}^{\text{avg-pro}}(M,p) \le e_{\mathbf{p}_1}^{\text{avg-pro}}(M,8/\pi^2) \le e_{\mathbf{p}_1}^{\text{wor-pro}}(M,8/\pi^2)$ is obvious and, applying Corollary 2.3.5, we get

$$e_{\mathbf{p}_1}^{\text{avg-pro}}\left(M, \frac{8}{/}\pi^2\right) \le \frac{3}{4}\,\frac{\pi}{M}.$$

As before denote $\sigma_a = (M/\pi)\arcsin\sqrt{a}$. Let $a = \frac{1}{2} + x$. We are interested in the behavior of $\sigma_{1/2+x}$ for $|x| < \frac{1}{2}$. Clearly $\sigma_{1/2} = \frac{1}{4}M$. Let $|x| < \frac{1}{2}$. By Taylor's theorem, we have

$$\sigma_{1/2+x} = \frac{M}{4} + \frac{M}{\pi}\frac{x}{2\sqrt{(1-\xi_x)\xi_x}} \qquad \text{for} \quad \xi_x \in (\tfrac{1}{2}, \tfrac{1}{2}+x)$$

and $2\sqrt{(1-\xi_x)\xi_x} \geq \sqrt{1-4x^2}$. Assume additionally that

$$\frac{M}{\pi}\frac{|x|}{\sqrt{1-4x^2}} \leq \frac{1}{4},$$

which is equivalent to assuming that

$$|x| \leq \frac{\pi}{(16M^2+4\pi^2)^{1/2}}.$$

Since M is divisible by 4, then $\lfloor \sigma_{1/2+x} \rfloor = \frac{1}{4}M$ for $x \geq 0$, and $\lceil \sigma_{1/2+x} \rceil = \frac{1}{4}M$ for $x \leq 0$. This yields

$$\min\left\{ \lceil \sigma_{1/2+x} \rceil - \sigma_{1/2+x}, \sigma_{1/2+x} - \lfloor \sigma_{1/2+x} \rfloor \right\} \leq \frac{M}{\pi}\frac{|x|}{\sqrt{1-4x^2}}. \tag{2.24}$$

We claim that

$$\lceil \sigma_{1/2+x} \rceil - \sigma_{1/2+x} \in [0, \tfrac{1}{4}] \cup [\tfrac{3}{4}, 1].$$

Indeed, for $x \leq 0$ we have

$$\lceil \sigma_{1/2+x} \rceil - \sigma_{1/2+x} = \frac{M}{\pi}\frac{|x|}{2\sqrt{(1-\xi_x)\xi_x}} \in [0, \tfrac{1}{4}],$$

and for $x \geq 0$ we have

$$\lceil \sigma_{1/2+x} \rceil - \sigma_{1/2+x} = 1 - \frac{M}{\pi}\frac{|x|}{2\sqrt{(1-\xi_x)\xi_x}} \in [1-\tfrac{1}{4}, 1] = [\tfrac{3}{4}, 1],$$

as claimed.

Let $a = \frac{1}{2} + x$. By the proof of Corollary 2.3.5, the error of the **QS** algorithm satisfies

$$|\bar{a} - a| \leq \frac{\pi}{M}\min\left\{ \lceil \sigma_{1/2+x} \rceil - \sigma_{1/2+x}, \sigma_{1/2+x} - \lfloor \sigma_{1/2+x} \rfloor \right\}$$

and by (2.24) we have

$$|\bar{a} - a| \leq \frac{|x|}{\sqrt{1-4x^2}}.$$

We split the sum that defines $e_{\mathbf{p}_1}^{\text{avg-pro}}(M, 8/\pi^2)$ into two sums. The first sum is for $f \in \mathbb{B}_N$ for which $a = a_f = \frac{1}{2} + x$ with $|x| \le \pi/(16M^2 + 4\pi^2)^{1/2}$ and the second sum is for f for which $a = a_f = \frac{1}{2} + x$ with $|x| > \pi/(16M^2 + 4\pi^2)^{1/2}$. We estimate the error of the **QS** algorithm by $|x|/\sqrt{1-4x^2}$ and by the worst-case error $3\pi/(4M)$ for the first and second sums, respectively. Hence we have

$$e_{\mathbf{p}_1}^{\text{avg-pro}}\left(M, \frac{8}{\pi^2}\right) \le \sum_{f\colon |a_f - 1/2| \le \pi/(16M^2+4\pi^2)^{1/2}} \mathbf{p}_1(f) \frac{|a_f - \frac{1}{2}|}{\sqrt{1 - 4(a_f - \frac{1}{2})^2}} + \sum_{f\colon ||a_f - 1/2| > \pi/(16M^2+4\pi^2)^{1/2}} \mathbf{p}_1(f) \frac{3}{4}\frac{\pi}{M}.$$

Since $a_f = k/N$ for some integer $k \in [0, N]$, we have

$$e_{\mathbf{p}_1}^{\text{avg-pro}}\left(M, \frac{8}{\pi^2}\right) \le \sum_{k\colon |k/N - 1/2| \le \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k} \frac{|k/N - \frac{1}{2}|}{\sqrt{1 - 4(k/N - \frac{1}{2})^2}} + \frac{3}{4}\frac{\pi}{M} \sum_{k\colon |k/N - 1/2| > \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k}.$$

Since $1 - 4(k/N - \frac{1}{2})^2 \ge 1 - \pi^2/(4M^2) \ge \frac{3}{4}$, the first sum can be estimated as

$$\sum_{k\colon |k/N - 1/2| \le \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k} \frac{|k/N - \frac{1}{2}|}{\sqrt{1 - 4(k/N - \frac{1}{2})^2}} \le \frac{2}{\sqrt{3}} \sum_{k\colon |k/N - 1/2| \le \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k} \left|\frac{1}{2} - \frac{k}{N}\right|.$$

The second sum can be estimated by

$$\frac{3}{4}\frac{\pi}{M} \sum_{k\colon |k/N - 1/2| > \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k} \frac{(16M^2 + 4\pi^2)^{1/2}}{\pi} \left|\frac{k}{N} - \frac{1}{2}\right| \le 3\sqrt{1 + \frac{\pi^2}{4M^2}} \sum_{k\colon |k/N - 1/2| > \pi/(16M^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k} \left|\frac{k}{N} - \frac{1}{2}\right|.$$

Adding the estimates of these two sums, we obtain

$$e_{\mathbf{p}_1}^{\text{avg-pro}}\left(M, \frac{8}{\pi^2}\right) \le 3\sqrt{1 + \frac{\pi^2}{4M^2}} \sum_{k=0}^{N} 2^{-N}\binom{N}{k} \left|\frac{1}{2} - \frac{k}{N}\right|.$$

The last sum is given by (2.22) and estimated by (2.23). Hence

$$e^{\text{avg-pro}}_{\mathbf{p}_1}\left(M, \frac{8}{\pi^2}\right) \le \sqrt{\frac{3}{2\pi}}\sqrt{1+\frac{\pi^2}{4M^2}}\frac{1}{\sqrt{N-1}}\, e^{1/(12(N-1))}$$

which completes the proof. □

In the next theorem we consider the case when M is not divisible by 4.

Theorem 2.3.11. *Assume that $M > 4$ is not divisible by 4, and let $p \in (1/2, 8/\pi^2]$. Then the average-probabilistic error of the* **QS** *algorithm with respect to the measure* $\mathbf{p}_1$ *satisfies*

$$e^{\text{avg-pro}}_{\mathbf{p}_1}(M,p) \ge \frac{\pi}{4M}\left(1-\frac{1}{M}-\frac{1}{\beta}\right)\left(1-2\exp\left(-\frac{N\pi^2}{(8\beta M)^2}\right)\right) \qquad \forall \beta > 1.$$

Proof. Let $M = 4M' + \tau$ for $\tau \in \{1,2,3\}$. Let $\sigma_a = (M/\pi)\arcsin\sqrt{a}$. As in the proof of Theorem 2.3.10, for $|x| < \frac{1}{2}$ we have

$$\sigma_{1/2+x} = \frac{M}{4} + \frac{M}{\pi}\frac{x}{2\sqrt{(1-\xi_x)\xi_x}} \qquad \text{for} \quad \xi_x \in (\tfrac{1}{2}, \tfrac{1}{2}+x)$$

and $2\sqrt{(1-\xi_x)\xi_x} \ge \sqrt{1-4x^2}$. Assume additionally that

$$\frac{M}{\pi}\frac{|x|}{\sqrt{1-4x^2}} \le \frac{1}{4\beta},$$

which is equivalent to assuming that

$$|x| \le \frac{\pi}{(16M^2\beta^2 + 4\pi^2)^{1/2}}. \tag{2.25}$$

Thus for x satisfying (2.25), we have

$$\sigma_{1/2+x} = M' + \frac{\tau}{4} + \frac{M}{\pi}\frac{x\,\theta(x)}{\sqrt{1-4x^2}} \quad \text{with} \quad \theta(x) \in [0,1],$$

and $\lfloor \sigma_{1/2+x} \rfloor = M'$ and $\lceil \sigma_{1/2+x} \rceil = M'+1$.

From the proof of Corollary 2.3.3 we have $\mu(\{\lceil \sigma_{a_f} \rceil, \lfloor \sigma_{a_f} \rfloor\}, f) \ge 8/\pi^2$. Since $\mu(A,f) \ge p > \frac{1}{2}$ then either $\lceil \sigma_{a_f} \rceil \in A$ or $\lfloor \sigma_{a_f} \rfloor \in A$. We then estimate

$$e^{\text{avg-pro}}_{\mathbf{p}_1}\left(M, \frac{8}{\pi^2}\right) \ge \sum_{f \in \mathbb{B}_N} \mathbf{p}_1(f) \min\left\{ \left|a_f - \bar{a}_f(\lfloor \sigma_{a_f} \rfloor)\right|, \left|a_f - \bar{a}_f(\lceil \sigma_{a_f} \rceil)\right| \right\}. \tag{2.26}$$

We now estimate the error of the **QS** algorithm for $f \in \mathbb{B}_N$ such that $a_f = \frac{1}{2} + x$ for x satisfying (2.25) and the outcome $j = \lceil \sigma_{a_f} \rceil = M'$ or $j = \lfloor \sigma_{a_f} \rfloor = M' + 1$. Denote the outcome by $M' + \kappa$ for $\kappa \in \{0, 1\}$. By Taylor's theorem we have

$$\sin^2\left(\frac{M'+\kappa}{M}\pi\right) = \sin^2\left(\frac{\pi}{4} + \frac{\pi}{M}(\kappa - \tfrac{1}{4}\tau)\right) = \frac{1}{2} + \sin(2\xi_{\kappa,\tau})\frac{\pi}{M}(\kappa - \tfrac{1}{4}\tau).$$

for $\xi_{\kappa,\tau} \in [\frac{1}{4}\pi, \frac{1}{4}\pi + (\pi/M)(\kappa - \frac{1}{4}\tau)]$. Since $\sin(t) \geq 2t/\pi$ for $t \in [0, \pi/2]$, we have $|\sin(2\xi_{\kappa,\tau})| \geq 1 - |4\kappa - \tau|/M$. Consider the error for the outcome $M' + \kappa$ and x satisfying (2.25). Then $|x| \leq \pi/(4\beta M)$ and the error can be estimated by

$$\begin{aligned}\left|\frac{1}{2} + x - \sin^2\left(\frac{M'+\kappa}{M}\pi\right)\right| &= \left|x - \sin(2\xi_{\kappa,\tau})\frac{\pi}{M}(\kappa - \tfrac{1}{4}\tau)\right| \\ &\geq \frac{\pi}{M}|\kappa - \tfrac{1}{4}\tau||\sin(2\xi_{\kappa,\tau})| - |x| \geq \frac{\pi}{4}\frac{|4\kappa-\tau|}{M}\left(1 - \frac{|4\kappa-\tau|}{M}\right) - \frac{\pi}{4\beta M}.\end{aligned}$$

Clearly, $|4\kappa - \tau| \in \{1, 2, 3\}$ and $|4\kappa - \tau|/M \in [1/M, 3/M]$. Then

$$|4\kappa - \tau|(1 - |4\kappa - \tau|/M)/M \geq (1 - 1/M)/M.$$

Therefore

$$\left|\frac{1}{2} + x - \sin^2\left(\frac{M'+\kappa}{M}\pi\right)\right| \geq \frac{\pi}{M}\left(\frac{1}{4}\left(1 - \frac{1}{M}\right) - \frac{1}{4\beta}\right) = \frac{\pi}{4M}\left(1 - \frac{1}{M} - \frac{1}{\beta}\right).$$

Hence, for f such that $a_f = \frac{1}{2} + x$ with x satisfying (2.25) we have

$$\min\left\{\left|a_f - \bar{a}_f\left(\lfloor \sigma_{a_f} \rfloor\right)\right|, \left|a_f - \bar{a}_f\left(\lceil \sigma_{a_f} \rceil\right)\right|\right\} \geq \frac{\pi}{4M}\left(1 - \frac{1}{M} - \frac{1}{\beta}\right). \tag{2.27}$$

We are now ready to estimate $e_{\mathbf{p}_1}^{\text{avg-pro}}(M, p)$. First, by (2.26), we have

$$\begin{aligned}e_{\mathbf{p}_1}^{\text{avg-pro}}(M, p) \geq &\sum_{f:\, |a_f - 1/2| \leq (16M^2\beta^2 + 4\pi^2)^{1/2}} \mathbf{p}_1(f) \\ &\times \min\left\{\left|a_f - \bar{a}_f\left(\lfloor \sigma_{a_f} \rfloor\right)\right|, \left|a_f - \bar{a}_f\left(\lceil \sigma_{a_f} \rceil\right)\right|\right\}.\end{aligned}$$

This, (2.27), and the Bernstein inequality, $\sum_{k:\,|k/N-1/2|>\varepsilon} 2^{-N}\binom{N}{k} \le 2e^{-N\varepsilon^2/4}$, yield

$$
\begin{aligned}
e^{\text{avg-pro}}_{\mathbf{p}_1}(M,p) &\ge \sum_{f:\,|a_f-1/2|\le\pi(16M^2\beta^2+4\pi^2)^{1/2}} \mathbf{p}_1(f)\,\frac{\pi}{4M}\left(1-\frac{1}{M}-\frac{1}{\beta}\right)\\
&= \frac{\pi}{4M}\left(1-\frac{1}{M}-\frac{1}{\beta}\right)\sum_{k:\,|k/N-1/2|\le\pi(16M^2\beta^2+4\pi^2)^{1/2}} 2^{-N}\binom{N}{k}\\
&\ge \frac{\pi}{4M}\left(1-\frac{1}{M}-\frac{1}{\beta}\right)\left(1-2\exp\left(-\frac{N\pi^2}{4(16M^2\beta^2-4)}\right)\right)\\
&\ge \frac{\pi}{4M}\left(1-\frac{1}{M}-\frac{1}{\beta}\right)\left(1-2\exp\left(-\frac{N\pi^2}{(8\beta M)^2}\right)\right),
\end{aligned}
$$

which completes the proof. □

Obviously, in the average-probabilistic setting, we should use the **QS** algorithm with M divisible by 4. Then Theorem 2.3.10 states that the error is of order $\min\{M^{-1}, N^{-1/2}\}$. Papageorgiou [36] proved that for any quantum algorithm that uses M quantum queries the error is bounded from below by $c\min\{M^{-1}, N^{-1/2}\}$ with probability $p \in (1/2, 8/\pi^2]$. Here, c is a positive number independent of M and N. Hence, the **QS** algorithm is optimal also in the average-probabilistic setting for the measure $\mathbf{p}_1$ as long as we use it with M divisible by 4.

We now turn to the measure $\mathbf{p}_2$. Clearly, the average-probabilistic error of the **QS** algorithm is bounded by its worst-probabilistic error, which is of order M^{-1} with probability $p \in (1/2, 8/\pi^2]$. It turns out, again due to a recent result of Papageorgiou [36] that this bound is the best possible, since any quantum algorithm that uses M quantum queries must have an error proportional at least to M^{-1}. Hence, the factor $N^{-1/2}$ that is present for the measure $\mathbf{p}_1$ does not appear for the measure $\mathbf{p}_2$, and the behavior of the **QS** algorithm is roughly the same in the worst- and average-probabilistic settings.

2.3.3 Worst-average error

The **QS** algorithm uses $M-1$ quantum queries. In this section, we restrict our consideration to the only interesting case, namely, when M is much smaller than N. We denote

$$s_{a_f} = \min\left\{ \lceil \sigma_{a_f} \rceil - \sigma_{a_f}, \sigma_{a_f} - \lfloor \sigma_{a_f} \rfloor \right\},$$

where, as before, $\sigma_{a_f} = (M/\pi)\arcsin\sqrt{a}$. Clearly, $s_{a_f} \in [0, \frac{1}{2}]$ and $s_{a_f} = 0$ iff σ_{a_f} is an integer. We shall usually drop the subscript f and denote $\sigma_a = \sigma_{a_f}$, $s_a = s_{a_f}$ when f is clear from the context. As in Section 1.2.1.1, let $\mu(\cdot, f)$ denote the measure on the set $\{0, \ldots, M-1\}$ of all possible outcomes of the **QS** algorithm. We also define

$$\mathcal{A}_M = \left\{ \sin^2\left(\frac{\pi j}{M}\right) : \ j = 0, 1, \ldots, M-1 \right\}$$

as the set of all possible outputs of the **QS** algorithm with $M-1$ queries. Let

$$\rho_f(\alpha) = \mu\left(\left\{ j \in \{0, 1, \ldots, M-1\} : \sin^2\left(\frac{\pi j}{M}\right) = \alpha \right\}, f\right) \quad \forall \alpha \in \mathcal{A}_M,$$

denote the probability of the output α. Note that $\alpha = \sin^2(\pi j/M) = \sin^2(\pi(M-j)/M)$. Hence, from (2.6), we clearly see that $\rho_f(\alpha) = p_f(j) + p_f(M-j)$ for $j \neq 0, M/2$.

For $q \in [1, \infty)$, we first analyze the local average error

$$e_q^{\text{avg}}(f, M) := \left(\sum_{j=0}^{M-1} p_f(j)\, |a_f - \bar{a}_f(j)|^q \right)^{1/q} = \left(\sum_{\alpha \in \mathcal{A}_M} \rho_f(\alpha) |a_f - \alpha|^q \right)^{1/q}, \qquad (2.28)$$

for a fixed function $f \in \mathbb{B}_N$.

2.3.3.1 Local average error

The local average error (2.28) for $q > 1$ is estimated in the following theorem.

Theorem 2.3.12. *Let $q \in (1, \infty)$. Denote $a = a_f$. If $\sigma_a \in \mathbb{Z}$, then $e_q^{avg}(f, M) = 0$.*

If $\sigma_a \notin \mathbb{Z}$*, then*

$$\left| e_q^{avg}(f,M)^q - \frac{\sin^2(\pi s_a)}{M\pi} \int_{\pi \overline{s}_a/M}^{\pi - \pi \underline{s}_a/M} \sin(x)^{q-2} \big| \sin(x + 2\theta_a) \big|^q \, dx \right| \le$$

$$(1 + 2(1-\delta_{q,2})) \frac{\pi^{q-1}\sin(\pi s_a)}{M^q} + \frac{\sin^2(\pi s_a)}{M^2} \left(2(1-\delta_{q,2}) + q \int_0^\pi \sin^{q-2}(x)\, dx \right), \quad (2.29)$$

with $\underline{s}_a = \lfloor \sigma_a \rfloor - \sigma_a$ *and* $\overline{s}_a = \sigma_a - \lceil \sigma_a \rceil$*.*[1]

Proof. If $\sigma_a \in \mathbb{Z}$ then it was shown in Section 2.3.2 that there exists $\alpha \in \mathcal{A}_M$ such that $\alpha = a_f$ and $\rho_f(\alpha) = \delta_{\alpha,a_f}$ for all $\alpha \in \mathcal{A}_M$. Then $e_q^{\text{avg}}(f,M) = 0$ as claimed.

Assume that $\sigma_a \notin \mathbb{Z}$. Using the closed form of of $p_f(j)$ as in (2.6), we rewrite (2.28) as

$$\left(e_q^{\text{avg}}(f,M)\right)^q = \sum_{j=0}^{M-1} \frac{\sin^2(M\theta_a)}{2M^2} \left(\left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^q + \left| \sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^q \right).$$

We have

$$\sum_{j=0}^{M-1} \left| \sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^q = \sum_{j=1}^{M} \left| \sin\left(\frac{\pi(M-j+\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(M-j-\sigma_a)}{M}\right)\right|^q.$$

Using the π-periodicity of $|\sin x|$, we see that the last sum is equal to

$$\sum_{j=1}^{M} \left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^q = \sum_{j=0}^{M-1} \left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^{q-2} \left| \sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^q.$$

---

[1] Note that the last integral is finite. This is obvious for $q \ge 2$. For $q \in (1,2)$, the only singularities are at the boundary points and are of the form x^{q-2} for x approaching 0. The function x^{q-2} is integrable since $q > 1$.

Therefore

$$e_q^{\mathrm{avg}}(f,M)^q = \frac{\sin^2(M\theta_a)}{M^2} S_{M,q}, \tag{2.30}$$

with

$$\begin{aligned} S_{M,q} = \sum_{j=0}^{M-1} \left| \sin\left(\frac{\pi(j-\sigma_a)}{M}\right)\right|^{q-2} \left|\sin\left(\frac{\pi(j+\sigma_a)}{M}\right)\right|^{q} \\ = \sum_{j=0}^{M-1} \left| \sin\left(\frac{\pi j}{M} - \theta_a\right)\right|^{q-2} \left|\sin\left(\frac{\pi j}{M} + \theta_a\right)\right|^{q}. \end{aligned}$$

We split $S_{M,q}$ as

$$S'_{M,q} = S_{M,q} - \left| \sin\left(\frac{\pi \left\lfloor \sigma_a \right\rfloor}{M} - \theta_a\right)\right|^{q-2} \left|\sin\left(\frac{\pi \left\lfloor \sigma_a \right\rfloor}{M} + \theta_a\right)\right|^{q}.$$

Observe that $(\pi/M)S'_{M,q}$ is the rectangle formula for approximating the integral

$$\int_{[0,\pi]\setminus[\pi\lfloor\sigma_a\rfloor/M,\pi\lceil\sigma_a\rceil/M]} \left|\sin(x-\theta_a)\right|^{q-2}\left|\sin(x+\theta_a)\right|^{q}\,dx.$$

The error of the rectangle quadrature for $k \in \mathbb{N}$ and an absolutely continuous function $f : [a,b] \to \mathbb{R}$ whose first derivative belongs to $L_1([a,b])$ satisfies

$$\left| \int_a^b f(x)\,dx - \frac{b-a}{k}\sum_{j=0}^{k-1} f\left(a + j\frac{b-a}{k}\right)\right| \le \frac{b-a}{k}\int_a^b \left|f'(x)\right|dx. \tag{2.31}$$

Defining $h(x) = \left|\sin(x-\theta_a)\right|^{q-2}\left|\sin(x+\theta_a)\right|^{q}$ and $D_a = [0,\pi]\setminus[\pi\left\lfloor\sigma_a\right\rfloor/M, \pi\left\lceil\sigma_a\right\rceil/M]$ and using the error formula (2.31) for the subintervals $[0,\pi\left\lfloor\sigma_a\right\rfloor/M)$ and $(\pi\left\lceil\sigma_a\right\rceil/M,\pi]$, we get

$$\left|\frac{\pi}{M}S'_{M,q} - \int_{D_a} h(x)\,dx\right| \le \frac{\pi}{M}\int_{D_a}\left|h'(x)\right|dx.$$

Define $H(x) = h(x+\theta_a) = \left|\sin(x)\right|^{q-2}\left|\sin(x+2\theta_a)\right|^{q}$ and $\Delta_a = [-\theta_a,\pi-\theta_a]\setminus[\pi(\lfloor\sigma_a\rfloor-\sigma_a)/M, \pi(\sigma_a-\lceil\sigma_a\rceil)/M]$. We have

$$\int_{D_a} h(x)\,dx = \int_{\Delta_a} H(x)\,dx, \qquad \int_{D_a}|h'(x)|\;dx = \int_{\Delta_a}|H'(x)|\;dx.$$

By the π-periodicity of the integrand H we have

$$\begin{aligned} \int_{\Delta_a} H(x)\,dx &= \int_{-\theta_a}^{-\pi\underline{s}_a/M} H(x)\,dx + \int_{\pi\overline{s}_a/M}^{\pi-\theta_a} H(x)\,dx \\ &= \int_{\pi-\theta_a}^{\pi-\pi\underline{s}_a/M} H(x)\,dx + \int_{\pi\overline{s}_a/M}^{\pi-\theta_a} H(x)\,dx = \int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} H(x)\,dx. \end{aligned}$$

Analogously,

$$\int_{\Delta_a} |H'(x)|\, dx = \int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} |H'(x)|\, dx.$$

For $x \in [\pi\overline{s}_a/M, \pi - \pi\underline{s}_a/M]$ the sine is positive and

$$|H'(x)| \leq |q-2| \sin^{q-3}(x) |\cos(x)| + q \sin^{q-2}(x).$$

It is easy to check that for $q \neq 2$ we have

$$\begin{aligned}
&\int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} |q-2| \sin^{q-3}(x) |\cos(x)|\, dx \\
&\qquad = |q-2| \bigg(\int_{\pi\overline{s}_a/M}^{\pi/2} \sin^{q-3}(x)\, d\sin(x) - \int_{\pi/2}^{\pi-\pi\underline{s}_a/M} \sin^{q-3}(x)\, d\sin(x) \bigg) \\
&\qquad\qquad = \frac{|q-2|}{q-2} \left(2 - \sin^{q-2}\left(\frac{\pi\overline{s}_a}{M}\right) - \sin^{q-2}\left(\pi - \frac{\pi\underline{s}_a}{M}\right)\right).
\end{aligned}$$

From this we get

$$\begin{aligned}
\int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} |H'(x)|\, dx \leq (1-\delta_{q,2}) \Bigg(2 + \sin^{q-2}\left(\frac{\pi\overline{s}_a}{M}\right) \\
+ \sin^{q-2}\left(\frac{\pi\underline{s}_a}{M}\right)\Bigg) + q \int_0^{\pi} \sin^{q-2}(x)\, dx.
\end{aligned}$$

We then finally get

$$\begin{aligned}
\left| \frac{\pi}{M} S_{M,q} - \int_{\pi\underline{s}_a/M}^{\pi-\pi\overline{s}_a/M} H(x)\, dx \right| \leq \frac{\pi}{M} \Bigg((1-\delta_{q,2}) \Bigg(2 + \sin^{q-2}\left(\frac{\pi\overline{s}_a}{M}\right) \\
+ \sin^{q-2}\left(\frac{\pi\underline{s}_a}{M}\right)\Bigg) + \sin^{q-2}\left(\frac{\pi\underline{s}_a}{M}\right) + q \int_0^{\pi} \sin^{q-2}(x)\, dx \Bigg).
\end{aligned}$$

Observe also that

$$\sin(\pi s_a) = \sin(\pi\underline{s}_a) = \sin(\pi\overline{s}_a).$$

Since $\sin(x)/[M\sin(x/M)] \le 1$ for $x \in (0,\pi]$, we get

$$\left|\frac{\pi\sin(\pi s_a)}{M} S_{M,q} - \sin(\pi s_a)\int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} H(x)\,dx\right|$$
$$\le \pi(1-\delta_{q,2})\left(\sin^{q-1}\left(\frac{\pi\overline{s}_a}{M}\right)+\sin^{q-1}\left(\frac{\pi\underline{s}_a}{M}\right)\right)+\pi\sin^{q-1}\left(\frac{\pi\underline{s}_a}{M}\right)$$
$$+\frac{\pi\sin(\pi s_a)}{M}\left(2(1-\delta_{q,2})+q\int_0^\pi \sin^{q-2}(x)\,dx\right).$$

Using $\sin(\pi\overline{s}_a/M) \le \pi/M$ we obtain

$$\left|\frac{\pi\sin(\pi s_a)}{M} S_{M,q} - \sin(\pi s_a)\int_{\pi\overline{s}_a/M}^{\pi-\pi\underline{s}_a/M} H(x)\,dx\right|$$
$$\le (1+2(1-\delta_{q,2}))\frac{\pi^q}{M^{q-1}}+\frac{\pi\sin(\pi s_a)}{M}\left(2(1-\delta_{q,2})+q\int_0^\pi \sin^{q-2}(x)\,dx\right).$$

Finally, since $\sin^2(M\theta_a)=\sin^2(\pi s_a)$, we complete the proof by using the estimate of $S_{M,q}$ in (2.30). $\square$

Theorem 2.3.12 implies the following corollary.

Corollary 2.3.13. *Let $q \in (1,\infty)$. If $\sigma_a \in \mathbb{Z}$, then $e_q^{avg}(f,M)=0$. If $\sigma_a \notin \mathbb{Z}$, then*

$$e_q^{avg}(f,M)=\frac{1}{M^{1/q}}\left[\frac{\sin^2(\pi s_a)}{\pi}\left(\int_0^\pi \sin^{q-2}(x)\big|\sin(x+2\theta_a)\big|^q\,dx + O\left(\frac{\sin(\pi s_a)}{M^{\min(1,q-1)}}\right)\right)\right]^{1/q}, \quad (2.32)$$

with $s_a \in (0,\frac{1}{2}]$, where the factor in the big O notation is independent of f from $\mathbb{B}_N$, and also independent of N.

We now consider the case $q=1$ and present estimates of $e_1^{\mathrm{avg}}(f,M)$ in the following lemma.

Lemma 2.3.14. *Let $a=a_f$. If $\sigma_a \in \mathbb{Z}$, then $e_1^{avg}(f,M)=0$. If $\sigma_a \notin \mathbb{Z}$, then*

$$\left|e_1^{avg}(f,M)-\frac{\sin^2(\pi s_a)\sin(2\theta_a)}{M}\,\Sigma_{M,a}\right| \le \frac{\sin^2(\pi s_a)}{M}|\cos(2\theta_a)|, \quad (2.33)$$

where $s_a \in (0, \frac{1}{2}]$, *and*

$$\Sigma_{M,a} = \frac{1}{M} \sum_{j=0}^{M-1} \left| \cot\left(\frac{\pi(j+s_a)}{M} \right) \right|.$$

Proof. The case $\sigma_a \in \mathbb{Z}$ can be proved as in Theorem 2.3.12. Assume that $\sigma_a \notin \mathbb{Z}$. Using (2.6), we get

$$e_1^{\text{avg}}(f,M) = \sum_{j=0}^{M-1} \frac{\sin^2(M\theta_a)}{2M^2} \left(\left| \frac{\sin(\pi(j+\sigma_a)/M)}{\sin(\pi(j-\sigma_a)/M)} \right| + \left| \frac{\sin(\pi(j-\sigma_a)/M)}{\sin(\pi(j+\sigma_a)/M)} \right| \right)$$

As in the proof of Theorem 2.3.12 we conclude that

$$e_1^{\text{avg}}(f,M) = \frac{\sin^2(M\theta_a)}{M^2} S_{M,1},$$

where

$$S_{M,1} = \sum_{j=0}^{M-1} \left| \frac{\sin(\pi(j+\sigma_a)/M)}{\sin(\pi(j-\sigma_a)/M)} \right| = \sum_{j=0}^{M-1} \left| \frac{\sin(\pi(j-\lceil\sigma_a\rceil+\overline{s}_a)/M+2\theta_a)}{\sin(\pi(j-\lceil\sigma_a\rceil+\overline{s}_a)/M)} \right|,$$

with $\overline{s}_a = \lceil\sigma_a\rceil - \sigma_a$. Changing the index j in the second sum to $j - \lceil\sigma_a\rceil$, and using periodicity of the sine, we get

$$S_{M,1} = \sum_{j=0}^{M-1} \left| \frac{\sin(\pi(j+\overline{s}_a)/M+2\theta_a)}{\sin(\pi(j+\overline{s}_a)/M)} \right|$$

and consequently

$$S_{M,1} = \sum_{j=0}^{M-1} \left| \cos(2\theta_a) + \sin(2\theta_a)\cot\left(\frac{\pi(j+\overline{s}_a)}{M} \right) \right|.$$

Using the triangle inequality twice, we obtain

$$\left| S_{M,1} - \sin(2\theta_a) \sum_{j=0}^{M-1} \left| \cot\left(\frac{\pi(j+\overline{s}_a)}{M} \right) \right| \right| \le M|\cos(2\theta_a)|.$$

Let $\underline{s}_a = \sigma_a - \lfloor\sigma_a\rfloor$. Observe that $\underline{s}_a = 1 - \overline{s}_a$. Since the cotangent is π-periodic and the function $|\cot(\pi(\cdot)/M)|$ is even, we get

$$\sum_{j=0}^{M-1} \left| \cot\left(\frac{\pi(j+\overline{s}_a)}{M} \right) \right| = \sum_{j=0}^{M-1} \left| \cot\left(\frac{\pi(j+\underline{s}_a)}{M} \right) \right| = M\,\Sigma_{M,a}.$$

This and

$$\sin^2(M\theta_a) = \sin^2(\pi\sigma_a) = \sin^2(\pi s_a)$$

yield (2.33), as claimed. □

From Lemma 2.3.14, we see that the sum $\Sigma_{M,a}$ is the most important part of the local average error $e_1^{\mathrm{avg}}(M,f)$. We now estimate $\Sigma_{M,a}$.

Lemma 2.3.15. *Assume that* $\sigma_a \notin \mathbb{Z}$ *and* $M \geq 3$. *Then*

$$\left| \Sigma_{M,a} - \frac{1}{M}\cot\left(\frac{\pi s_a}{M}\right) - \frac{1}{M}\left|\cot\left(\frac{\pi(M-1+s_a)}{M}\right)\right| \right.$$
$$\left. - \frac{1}{\pi}\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} |\cot x|\,dx \right| \leq \frac{1}{\pi M}\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} \frac{1}{\sin^2 x}\,dx. \quad (2.34)$$

Proof. This can be shown by using the error formula for rectangle quadratures (2.31). Note that $\pi\Sigma_{M,a} - (\pi/M)\cot(\pi s_a/M) - (\pi/M)|\cot(\pi(M-1+s_a)/M)|$ is the rectangle quadrature for the integral $\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} |\cot x|\,dx$ with $k = M-2 \geq 1$. We then obtain (2.34) by using (2.31). □

We now present the final estimate on the local average error $e_1^{\mathrm{avg}}(f,M)$.

Theorem 2.3.16. *Assume that* $f \in \mathbb{B}_N$ *and* $a = a_f$. *For* $M \geq 3$, *the average error of the* **QS** *algorithm for the function* f *satisfies*

$$\left| e_1^{avg}(f,M) - \frac{2\sin^2(\pi s_a)\sin(2\theta_a)}{\pi}\,\frac{\ln M}{M} \right| \leq \frac{3\pi + 2 + \ln(\pi^2)}{M\pi}\,\sin(\pi s_a). \qquad (2.35)$$

Proof. For $\sigma_a \in \mathbb{Z}$ we have $s_a = 0$, so that (2.35) holds since $e_1^{\mathrm{avg}}(f,M) = 0$ by [30]. Assume that $\sigma_a \notin \mathbb{Z}$. From Lemmas 2.3.14 and 2.3.15 we have

$$\left| e_1^{\mathrm{avg}}(f,M) - \frac{\sin^2(\pi s_a)\sin(2\theta_a)}{\pi M}\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} |\cot x|\,dx \right|$$
$$\leq \frac{\sin^2(\pi s_a)}{M}\left[\frac{\sin(2\theta_a)}{M}\left(\cot\left(\frac{\pi s_a}{M}\right) + \left|\cot\left(\frac{\pi(M-1+s_a)}{M}\right)\right| \right.\right.$$
$$\left.\left. + \frac{1}{\pi}\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} \frac{1}{\sin^2 x}\,dx \right) + |\cos(2\theta_a)| \right].$$

Observe that

$$\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} |\cot x|\,dx = \ln\left(\sin^{-1}\left(\frac{\pi(1+s_a)}{M}\right)\sin^{-1}\left(\frac{\pi(1-s_a)}{M}\right)\right),$$

$$\left|\cot\left(\frac{\pi(M-1+s_a)}{M}\right)\right| = \cot\left(\frac{\pi(1-s_a)}{M}\right) \le \cot\left(\frac{\pi s_a}{M}\right),$$

$$\begin{aligned}\int_{\pi(1+s_a)/M}^{\pi(M-1+s_a)/M} \frac{1}{\sin^2 x}\,dx &= \cot\left(\frac{\pi(1-s_a)}{M}\right) + \cot\left(\frac{\pi(1+s_a)}{M}\right)\\ &\le 2\cot\left(\frac{\pi s_a}{M}\right).\end{aligned}$$

The four formulas above yield

$$\begin{aligned}&\left|e_1^{\mathrm{avg}}(f,M) - \frac{\sin^2(\pi s_a)\sin(2\theta_a)}{\pi M}\ln\left(\sin^{-1}\left(\frac{\pi(1+s_a)}{M}\right)\sin^{-1}\left(\frac{\pi(1-s_a)}{M}\right)\right)\right|\\ &\qquad\le \frac{\sin^2(\pi s_a)}{M}\left(\frac{(2+2/\pi)\sin(2\theta_a)}{M}\cot\left(\frac{\pi s_a}{M}\right) + |\cos(2\theta_a)|\right).\end{aligned}$$

Observe that $\sin(\pi s_a)/[M\sin(\pi s_a/M)] \le 1$ since $s_a \in (0, \frac{1}{2}]$. This and the obvious estimates of sine and cosine yield

$$\begin{aligned}&\left|e_1^{\mathrm{avg}}(f,M) - \frac{\sin^2(\pi s_a)\sin(2\theta_a)}{\pi M}\ln\left(\sin^{-1}\left(\frac{\pi(1+s_a)}{M}\right)\sin^{-1}\left(\frac{\pi(1-s_a)}{M}\right)\right)\right|\\ &\qquad\le \left(3+\frac{2}{\pi}\right)\frac{\sin(\pi s_a)}{M}. \qquad (2.36)\end{aligned}$$

Consider now the left hand side of (2.36). Remembering that $M \ge 3$, and since $2x/\pi \le \sin x \le x$ for $x \in [0, \pi/2]$, we get

$$\left|\ln\left(\sin^{-1}\left(\frac{\pi(1+s_a)}{M}\right)\sin^{-1}\left(\frac{\pi(1-s_a)}{M}\right)\right) - 2\ln M\right| \le \ln(\pi^2). \qquad (2.37)$$

Thus by (2.36) and (2.37), we get the final estimate (2.35). □

From Corollary 2.3.12 and Theorem 2.3.16, we get sharp estimates on the worst-average error of the **QS** algorithm.

Theorem 2.3.17. *Let $M \ge 3$. Then the worst-average error of the* **QS** *algorithm satisfies the following bounds.*

- *For* $q \in (1,\infty)$,

$$e_q^{\textit{wor-avg}}(M) \le \frac{1}{M^{1/q}}\left(\frac{1}{\pi}\int_0^{\pi} \sin^{q-2}(x)\,dx\right)^{1/q}\big(1+o(1)\big). \tag{2.38}$$

The last estimate is sharp, i.e.,

$$e_q^{\textit{wor-avg}}(M) = \Theta\left(\frac{1}{M^{1/q}}\right). \tag{2.39}$$

In particular, for $M-2$ *divisible by* 4 *we have*

$$e_q^{\textit{wor-avg}}(M) \ge \frac{1}{M^{1/q}}\left(\frac{1}{\pi}\int_0^{\pi} \sin^{q-2}(x)\,|\cos(x)|^q\,dx\right)^{1/q}\big(1+o(1)\big), \tag{2.40}$$

and the ratio of the integrals in (2.38) and (2.40) are approximately 1 *for* q *close to* 1.

- *For* $q=1$,

$$e_1^{\textit{wor-avg}}(M) \le \frac{2}{\pi}\,\frac{\ln M}{M} + \frac{3\pi+2+\ln(\pi^2)}{M\pi}. \tag{2.41}$$

This estimate is sharp, i.e.,

$$e_1^{\textit{wor-avg}}(M) = \Theta(M^{-1}\ln M). \tag{2.42}$$

In particular, for $M-2$ *divisible by* 4 *we have*

$$e_1^{\textit{wor-avg}}(M) \ge \frac{2}{\pi}\,\frac{\ln M}{M} - \frac{3\pi+2+\ln(\pi^2)}{M\pi}.$$

Proof. Consider first the case $q \in (1,\infty)$. By Corollary 2.3.13 we have

$$e_q^{\mathrm{avg}}(f,M) \;\le\; \frac{1}{M^{1/q}}\left(\frac{1}{\pi}\int_0^{\pi} \sin^{q-2}(x)\,dx\right)^{1/q}\big(1+o(1)\big) \qquad \forall f \in \mathbb{B}_N,$$

where $o(1)$ is independent of f. This yields (2.38).

The estimate (2.38) is sharp since if we let f be a Boolean function such that $s_{a_f} \approx \frac{1}{2}$, then (2.32) yields (2.39). In particular, for $M = 4k+2$ and $a_f = 1/2$ we have $\theta_{a_f} = \pi/4$, $\sigma_{a_f} = M/4 = k+1/2$ and $s_a = \frac{1}{2}$. Therefore

$$e_q^{\mathrm{avg}}(f,M) = \frac{1}{M^{1/q}}\left(\frac{1}{\pi}\int_0^{\pi} \sin^{q-2}(x)\,|\cos(x)|^q\,dx\right)^{1/q}\big(1+o(1)\big),$$

which proves (2.40). For q close to 1, the value of $\int_0^\pi \sin^{q-2}(x)dx$ is mostly due to the values of $\sin^{q-2}(x)$ close to 0 and π. Since $|\cos(x)|^q$ is then approximately equal to 1, the ratio of the upper and lower bound integrals is about 1.

For $q = 1$, the estimate (2.41) follows directly from Theorem 2.3.16. To prove (2.42) it is enough to choose a Boolean f for which the numbers

$$\sin^2(\pi s_{a_f}) \sin(2\theta_{a_f}) \;=\; \sin^2(M\theta_{a_f}) \sin(2\theta_{a_f})$$

are uniformly (in M) separated from 0, see Theorem 2.3.16. More precisely, since a_f can take any value k/N for $k = 0, 1, \ldots, N$, we take a Boolean function f such that $|a_f - \sin^2(\pi/4 + \pi/(5M))| \leq 1/(2N)$. For sufficiently large N, we have $\theta_{a_f} \approx \pi/4 + \pi/(5M)$. For large $M = 4k + \beta$ with $\beta \in \{0, 1, 2, 3\}$, we then have

$$\sin^2(M\theta_{a_f}) \sin(2\theta_{a_f}) \;\approx\; \sin^2\left(\frac{4+5\beta}{20}\pi\right) \sin\left(\frac{1}{2}\pi + \frac{1}{2.5M}\pi\right) \;>\; c > 0,$$

for some c independent of M.

In particular, for $M-2$ divisible by 4 we take $N > M$ and a Boolean function $f \in \mathbb{B}_N$ with $a_f = 1/2$. Then

$$s_a = \tfrac{1}{2} \quad \text{and} \quad \sin^2(\pi s_a) \sin(2\theta_a) \;=\; 1,$$

which leads the last estimate of Theorem 2.3.17. □

2.3.3.2 Quantum summation algorithm with repetitions

The success probability of the **QS** algorithm is increased by repeating it a number of times and taking the median of the outputs as the final output, see e.g., [15]. We show in this section that this procedure improves the worst-average error estimate.

We perform $2n + 1$ repetitions of the **QS** algorithm for some $n \in \{0, 1, \ldots\}$. We obtain $\sin^2(\pi j_1/M), \sin^2(\pi j_2/M), \ldots, \sin^2(\pi j_{2n+1}/M)$ and let $\bar{a}_{n,f}$ be the median of the outputs obtained, i.e., the $(n+1)$st number in the ordered sequence. Let, as before, $\mathcal{A}_M = \{\sin^2(\pi j/M) : j = 0, 1, \ldots, M-1\}$. For $\alpha \in \mathcal{A}_M$, let $\rho_{n,f}(\alpha)$ be the probability that the median $\bar{a}_{n,f}$ is equal to α. This probability depends on the distribution function F_f of the original outputs from $\mathcal{A}_M$, which is defined as

$$F_f(\alpha) = \begin{cases} \sum_{\alpha' \in \mathcal{A}_M, \alpha' < \alpha} \rho_f(\alpha') & \text{for } \alpha > 0, \\ 0 & \text{for } \alpha = 0. \end{cases}$$

It is known, see [41] p. 410, that the distribution of the median $\bar{a}_{n,f}$ is of the form

$$\rho_{n,f}(\alpha) = (2n+1)\binom{2n}{n} \int_{F_f(\alpha)}^{F_f(\alpha)+\rho_f(\alpha)} t^n(1-t)^n \, dt, \quad \forall \alpha \in \mathcal{A}_M. \tag{2.43}$$

We are now ready to estimate the worst-average error

$$e_{q,n}^{\text{wor-avg}}(M) = \sup_{N>M} \max_{f \in \mathbb{B}_N} \left(\sum_{\alpha \in \mathcal{A}_M} \rho_{n,f}(\alpha) |a_f - \alpha|^q \right)^{1/q}, \qquad q \in [1, \infty)$$

of the **QS** algorithm with $2n+1$ repetitions.

We estimate $e_{q,n}^{\text{wor-avg}}(M)$ by using Theorem 12 of [4] which states that the **QS** algorithm with M queries computes $\bar{a}_f$ such that

$$|a_f - \bar{a}_f| \ \geq \ c_1 \frac{k}{M} \quad \text{with probability at most} \quad \frac{c_2}{k}$$

for any positive integer k, Here c_1 and c_2 are absolute constants and f is any Boolean function from $\mathbb{B}_N$. If

$$|a_f - \bar{a}_{n,f}| \ \geq \ c_1 \frac{k}{M},$$

then for at least n outcomes $\bar{a}_f(j_1), \dots, \bar{a}_f(j_n)$ we must have

$$|a_f - \bar{a}_f(j_l)| \ \geq \ c_1 \frac{k}{M} \quad \text{for } l = 1, \dots, n.$$

But the probability that this occurs is bounded by

$$\binom{2n+1}{n} \left(\frac{c_2}{k}\right)^n.$$

It follows then that

$$\text{Prob}\{\, |a_f - \bar{a}_{n,f}| \ \geq \ c_1 k/M \,\} \ \leq \ c\, k^{-n},$$

where c which may depend on n. We now use the standard summation by parts. Define

$$p_k \;=\; \text{Prob}\{\, c_1(k-1)/M \;\le\; |a_f - \bar{a}_{n,f}| \;<\; c_1 k/M\}.$$

Then, by the estimate above we get,

$$\sum_{k>l} p_k \;\le\; c\, l^{-n} \qquad \forall l \in \mathbb{N}.$$

Therefore

$$\begin{aligned} e^{\text{wor-avg}}_{q,n}(M)^q &\le \sum_{k=1}^{\infty} p_k (c_1 k/M)^q \;=\; (c_1/M)^q \sum_{k=1}^{\infty} p_k \sum_{l=1}^{k} (l^q - (l-1)^q) \\ &= (c_1/M)^q \sum_{l=1}^{\infty} (l^q - (l-1)^q) \sum_{k=l}^{\infty} p_k \le c M^{-q} \sum_{l=1}^{\infty} l^{q-1-n} \;\le\; c\, M^{-q} \end{aligned}$$

for $n > q$, with the number $c = c_{q,n}$ depending only on q and n. In fact, taking $n = \lceil q \rceil + 1$ it is easy to check that $c_{q,n}$ is a single exponential function of q. Hence, by taking the qth root we have

$$e^{\text{wor-avg}}_{q,n}(M) \le c_{q,n}^{1/q} M^{-1},$$

with $c_{q,n}^{1/q}$ of order 1. Therefore we have proved the following theorem.

Theorem 2.3.18. *The worst-average error of the median of* $2(\lceil q \rceil + 1) + 1$ *repetitions of the* **QS** *algorithm with* M *quantum queries satisfies*

$$e^{\textit{wor-avg}}_{q,\lceil q \rceil + 1}(M) \;=\; O(M^{-1}),$$

with an absolute constant in the big O *notation independent of* q *and* M*.*

The essence of Theorem 2.3.18 is that the number of repetitions of the **QS** algorithm is *independent* of M and depends only linearly on q. Still, it permits us to dramatically improve the worst-average error of the **QS** algorithm. As we already mentioned in the introduction, the bound of order M^{-1} is a lower bound on the worst-average error of any quantum algorithm. Hence, the **QS** algorithm with repetitions is also optimal in the worst-average setting.

2.4 Simulation

In this section we show how the quantum Boolean summation algorithm for the class $\mathbb{B}_N$ can be simulated on a classical computer. From the previous sections we see that the core problem is the computation of amplitudes of the final state since they are needed to compute the probabilities of all possible outcomes. We present a MATLAB procedure `QSsimul` that computes all amplitudes of the final state before the measurement. The cost of `QSsimul` is of order $(N/\varepsilon)\log_2(1/\varepsilon)$.

2.4.1 QS algorithm simulation

Our simulation computes all amplitudes of the final state of the **QS** algorithm for a Boolean function $f \in \mathbb{B}_N$ and given error $\varepsilon \in (0,1)$.

First, based on the estimate (2.17), we compute the number m of qubits required to provide a desired accuracy with probability at least $8/\pi^2$. During the computation we use an $N \times M$ matrix $[a^i_{j,k}]_{j=0,k=0}^{M-1,N-1}$, where $M = 2^m$, with MN coefficients of the quantum states $|\eta_1\rangle$, $|\eta_2\rangle$ and $|\eta_3\rangle$ as defined in Section 2.2. The coefficients of the states $|\eta_i\rangle$ are defined with respect to the computational basis $\{|j\rangle|k\rangle\}_{j=0,k=0}^{M-1,N-1}$.

Clearly for $i = 1$ we have $a^1_{j,k} = (MN)^{-1/2}$. To obtain the state $|\eta_2\rangle$ we need to apply the Grover iterate operator $\Lambda_m(Q_f)$. We now show how to do this efficiently. We have

$$|\eta_2\rangle = \Lambda_m(Q)W_{m+n}|0\rangle = \frac{1}{\sqrt{MN}}\Lambda_m(Q)\sum_{j=0}^{M-1}|j\rangle\sum_{k=0}^{N-1}|k\rangle = \frac{1}{\sqrt{MN}}\sum_{j=0}^{M-1}|j\rangle\, Q^j\sum_{k=0}^{N-1}|k\rangle.$$

Observe that

$$\begin{aligned} Q\sum_{k=0}^{N-1}a_k|k\rangle &= \sum_{k=0}^{N-1} -a_kW_nS_0W_nS_f|k\rangle = \sum_{k=0}^{N-1} -(-1)^{f(k)}a_kW_n(I-2|0\rangle\langle 0|)W_n|k\rangle \\ &= \sum_{k=0}^{N-1} -(-1)^{f(k)}a_k|k\rangle + \frac{2}{\sqrt{N}}\sum_{k=0}^{N-1}a_k(-1)^{f(k)}W_n|0\rangle \\ &= \sum_{k=0}^{N-1}\left(\frac{2}{N}\sum_{j=0}^{N-1}a_j(-1)^{f(j)} - (-1)^{f(k)}a_k\right)|k\rangle. \end{aligned}$$

Thus, the coefficients $a^2_{j,k}$ of the state $|\eta_2\rangle$ satisfy the following recursive formula

$$a^2_{j,k} = \begin{cases} (MN)^{-1/2} & j = 0, \\ \dfrac{2}{N} \sum_{p=0}^{N-1} a^2_{j-1,p}(-1)^{f(p)} - (-1)^{f(k)} a^2_{j-1,k}, & j = 1, \dots, M-1, \end{cases} \tag{2.44}$$

for $k = 0, \dots, N-1$.

To obtain the state $|\eta_3\rangle$ we need to apply the inverse quantum Fourier transform $F_m^{-1} \otimes I$. Hence

$$|\eta_3\rangle = (F_m^{-1} \otimes I) \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} a^2_{j,k} |j\rangle|k\rangle = \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \sum_{p=0}^{M-1} a^2_{p,k} e^{-2\pi i p j/M} |j\rangle|k\rangle,$$

which yields the final form of the coefficients as

$$a^3_{j,k} = \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} a^2_{p,k} e^{-2\pi i p j/M}.$$

We see that the rows of the matrix $[a^3_{j,k}]$ are discrete Fourier transforms of the respective rows of the matrix $[a^2_{j,k}]$.

It is easy to see that the costs of the steps computing the states $|\eta_1\rangle$ and $|\eta_2\rangle$ are of order MN. We can compute the state $|\eta_3\rangle$ by using the Fast Fourier Transform algorithm with the cost of order $MNm = MN \log_2 M$. We see then that the total cost of our simulation is of order of the cost of computing the state $|\eta_3\rangle$. Assuming $m = \lceil \log_2(\pi/\varepsilon) \rceil$), see Corollary 2.3.9, the total cost is of order $(N/\varepsilon) \log_2 \varepsilon^{-1}$.

The simulation algorithm outlined above has been coded in GNU Octave, see [10], and can be easily ported to MATLAB.

The program `QSsimul` computes all the coefficients of the final state of the **QS** algorithm, stores numerical results and creates a PostScript file containing graphs of the exact value of the sum, the error bounds, possible outputs and their probabilities.

QSsimul.m

```
function [outputs, probabs, sp, pmax, avg, qubits]=QSsimul(n, eps, f)
#[outputs, probabs, sp, pmax, avg, qubits]= QSimul( n, eps, f )
#
#INPUT:
#n - number of qubits for coding the domain of the Boolean function f,
#    i.e., the cardinality is 2^n
#eps - the desired accuracy 0 < eps < 1
# f - Boolean function {0, .. 2^n -1} -> {0,1}
#OUTPUT:
#outputs - vector Mx1, all possible outputs of the QS algorithm
#probabs - vector Mx1, corresponding probabilities
#sp - the best estimate
#pmax - the probability of obtaining an estimate with  accuracy eps
#avg - exact value
#qubits - the number of qubits in the second register
#The visualization of the final state is in the file out.ps

  qubits=ceil(log2(pi/eps));
  M=2^qubits;
  N=2^n;
  Reg=ones(M, N);
  outputs=zeros(M,1);
  avg=0;
  i=0;

  fo=zeros(N,1);
  for i=1:N
    ff=feval(f,i-1);
    if ff==1
      avg++;
    end
    fo(i)=(-1)^ff;
  end

  avg/=N;

  for i=2:M
    sum=(2/N)*(Reg(i-1,:)*fo);
    for j=1:N
      Reg(i,j)=sum-Reg(i-1, j)*fo(j);
    end
  end

  Reg=1/(M*sqrt(N))*fft(Reg);

  p=0;
  sp=0;

  probabs=zeros(M,1);
  for l=1:M
    for ll=1:N
      probabs(l)=probabs(l)+abs(Reg(l,ll))^2;
    end
    outputs(l)=(sin(pi*(l-1)/M))^2;
    if (p < probabs(l))
      p = probabs(l);
    sp = l-1;
  end
end

sp=outputs(sp+1);
pmax=0;

for l=1:M
  if (abs(outputs(l)-avg)< eps)
    pmax=pmax+ probabs(l);
  end
end

data=[(linspace(0, M-1, M))',outputs, probabs, avg*ones(M,1)];
data= [data, sp*ones(i,1), (avg-eps)*ones(i,1), (avg+eps)*ones(i,1)];

gset output "out1.eps"
gset terminal postscript eps
gset style line 1 lt 1
gset style line 2 lt 2
gset style line 3 lt 3
gset style line 4 lt 4
gset key outside bottom
gset xlabel "States | j >             (Total # States = M)"
gset ylabel "Value of the sum"
gplot [0:M-1] [0:1] data using 1:2 title "Possible outputs" \
                                                with points,\
                    data using 1:3 title "Pr(state | j >)" \
                                                with boxes 1,\
                      data using 1:4 title "True value" \
                                           with lines 1,\
                      data using 1:5 title "Estimated value" \
                                                with lines 2,\
                      data using 1:6 title "Error bounds" \
                                           with lines 3,\
                      data using 1:7 title "" with lines 3
end
```

The results of running `QSsimul` for three different functions f are presented in Figures 2.3, 2.4, 2.5.

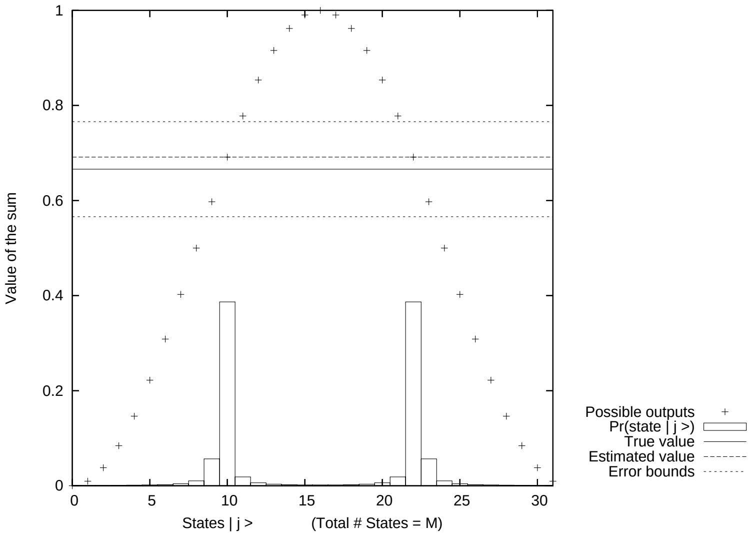

Figure 2.3: `QSsimul` output for $N = 2^{10}$, $\varepsilon = 0.1$, $M = 32$ and $f(k) = 1$ for k not divisible by 3 and $f(k) = 0$ otherwise.

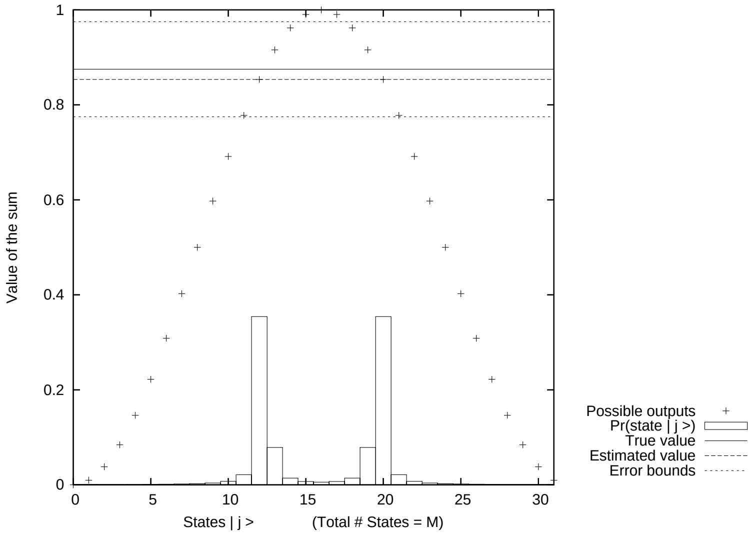

Figure 2.4: `QSsimul` output for $N = 2^{10}$, $\varepsilon = 0.1$, $M = 32$ and $f(k) = 1$ for k not divisible by 8 and $f(k) = 0$ otherwise.

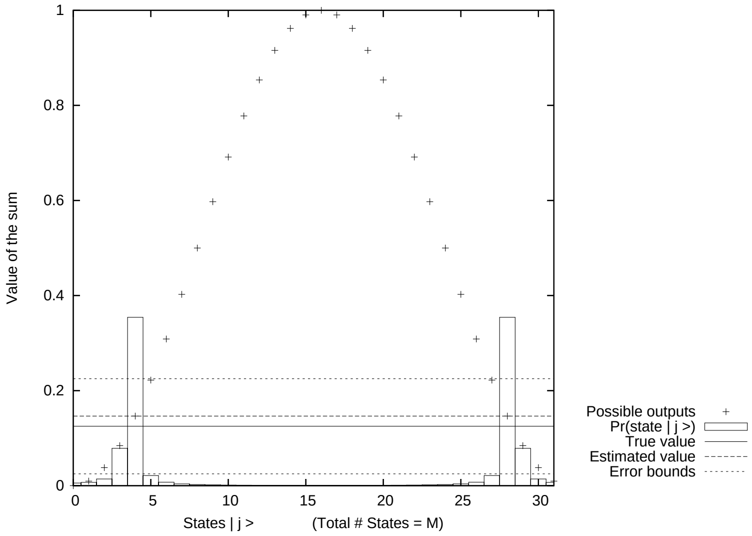

Figure 2.5: `QSsimul` output for $N = 2^{10}$, $\varepsilon = 0.1$, $M = 32$ and $f(k) = 1$ for k divisible by 8 and $f(k) = 0$ otherwise.

2.4.2 Computation of the final state distribution

Suppose that we want to know the distribution of the final state of the **QS** algorithm without computing all amplitudes of the final state. This problem can be solved much faster than the full simulation. First we need to compute the arithmetic mean of a Boolean function with cost of order N. Then, from (2.12) we can compute the probability the final outcome $|k\rangle$. Thus the cost of computing the final state distribution is clearly of order $M+N$, which is of order $\varepsilon^{-1}+N$ for the desired accuracy ε. Hence, we can save a factor of $\varepsilon^{-1}\log_2\varepsilon^{-1}$ over the cost of simulation.

2.5 Conclusions

The results of this chapter show the robustness of the **QS** algorithm. Its optimality in the the worst-probabilistic error setting was extended to two more error settings—average-probabilistic and worst-average.

The average-probabilistic error criterion is weaker than the worst-probabilistic one. The difference is that the probabilistic error is considered on the average with respect to the class $\mathbb{B}_N$ of input Boolean functions. We consider two measures on the class $\mathbb{B}_N$. The first measure is uniform on Boolean functions, while the second one is uniform on arithmetic means of Boolean functions. We show that for the first measure, the **QS** algorithm retains its optimality for a certain choice of its parameters. For the second measure, the average-probabilistic error is essentially of the same order as the worst-probabilistic one, so weakening the error criterion does not yield any essential cost gain over the worst-probabilistic error setting.

In the worst-average setting the **QS** algorithm with repetitions is optimal. This shows its superiority over classical randomized algorithms. Indeed, the worst-average error criterion with respect to L_2 norm is analogous to the usual error criterion by which the error of classical randomized algorithms is considered, among them the Monte Carlo algorithm. We recall from [34] that the complexity of the Boolean summation in the classical randomized setting is of order ε^{-2}, while for the **QS** algorithm with repetitions we have the optimal cost of order ε^{-1}. Furthermore, the worst-average error is a reasonable choice for algorithms for which we cannot compute the actual a posteriori error, so the result of an algorithm cannot be verified. The reason for this is that the worst-average error takes into account all the outcomes of an algorithm, while the worst-probabilistic error deals only with highly probable outcomes. Finally, the worst-average error criterion is stronger than the worst-probabilistic one, which can be easily proved by using Chebyshev's inequality.

Chapter 3

Multivariate Feynman-Kac path integration

3.1 Introduction

In this chapter, we analyze the multivariate Feynman-Kac path integration problem. Although we are mainly interested in the quantum setting we also include the worst-case deterministic and randomized settings, which will allow us to compare the results for all three settings.

We recall that multivariate Feynman-Kac path integrals are path integrals over the space of continuous functions from $\mathbb{R}_+$ to $\mathbb{R}^d$, equipped with a Wiener measure. The multivariate Feynman-Kac formula is the solution of the initial value problem

$$\frac{\partial z}{\partial t}(\mathbf{u},t) = \tfrac{1}{2}\Delta z(\mathbf{u},t) + V(\mathbf{u})z(\mathbf{u},t) \qquad \text{for } (\mathbf{u},t) \in \mathbb{R}^d \times (0,\infty), \tag{3.1}$$

$$z(\mathbf{u},0) = v(\mathbf{u}) \tag{3.2}$$

for the diffusion (heat) equation. Here $v, V : \mathbb{R}^d \to \mathbb{R}$ are the initial value function and the potential function. As usual, Δ denotes the Laplacian. The solution z of (3.1) and (3.2) is given by the famous Feynman-Kac formula

$$z(\mathbf{u},t) = \int_{\mathcal{C}} v(\mathbf{x}(t)+\mathbf{u}) \exp\left(\int_0^t V(\mathbf{x}(s)+\mathbf{u})\, ds\right) w(d\mathbf{x}). \tag{3.3}$$

Here, $\mathcal{C}$ is the set of continuous functions $\mathbf{x} : \mathbb{R}_+ \to \mathbb{R}^d$ such that $\mathbf{x}(0) = \mathbf{0}$. The path integral (3.3) is taken with respect to the d-dimensional Wiener measure w, see [24, 40]. Obviously, (3.3) only holds for functions v and V for which the path integral exists. In what follows, we assume that the functions v and V belong to a class F for which (3.3) exists. This class is precisely defined in Section 3.3.

Various computational algorithms, mostly randomized, have been developed for the univariate case $d = 1$, where the Feynman-Kac path integral is the solution of the diffusion equation with one space variable. A novel approach for the univariate case $d = 1$ was proposed in [39], where a new deterministic algorithm based on L_2-approximation was constructed and the complexity of the univariate case was studied. Those results were modified and generalized to the multivariate case in [26] and then improved in [29]. The multivariate algorithm retained the structure of the algorithm from [39] and is based on uniform approximation. We briefly discuss the worst-case setting in Sections 3.2.1, 3.5.2 and 3.6.1.

As we shall see, the quantum setting for multivariate Feynman-Kac path integration is strongly related to the randomized setting. We present algorithms that compute approximations of multivariate Feynman-Kac path integrals in both these settings. Both algorithms are based on uniform approximation, similarly to the deterministic algorithm from [29]. We analyze the (informational) costs of these algorithms, i.e., the numbers of function evaluations and/or quantum queries used to compute approximations with a given error bound. Finally we relate these costs to the complexity of multivariate Feynman-Kac path integration in the randomized and quantum settings. As in [29, 39], the complexity is bounded from below by the complexity of multivariate weighted integration. The upper bounds are provided by the costs of the algorithms presented in this chapter. As we shall see, the power of randomization or quantum computation yields a substantial improvement over the worst case complexity.

We now discuss the complexity results presented later in this chapter. We know that deterministic algorithms for the problem of approximating multivariate Feynman-Kac path integrals with the worst case error assurance are highly inefficient for large d, see [29]. This is caused by the *provable* curse of dimensionality of this problem. More precisely, the cost of computing an ε-approximation by any deterministic algorithm is at least of order $\varepsilon^{-d/r}$. Here, r measures the smoothness of the initial value and potential functions, e.g., we consider the class F of functions which are r times continuously differentiable. Clearly, the cost depends exponentially on d.

Switching to the randomized setting is one way to vanquish the curse of dimensionality. In this setting the path integral is approximated by a multivariate integral, and then this integral is approximately evaluated by a randomized algorithm, e.g., by the celebrated Monte Carlo algorithm, see, e.g., [5, 14], also [11]. This yields a cost of order $(1/\varepsilon)^2$, so that the dependence on d disappears. As we shall see, the exponent of $1/\varepsilon$ can be improved at the expense of introducing a dependence on d. More precisely, for positive r we construct an optimal algorithm with cost of order $\varepsilon^{-2/(1+2r/d)}$, see Section 3.6.2.

The use of a quantum computer yields even greater improvement. Let us measure the cost of an algorithm by the number of queries and function evaluations it uses. We show in Section 3.6.3 that an optimal quantum algorithm computes an ε-approximation with cost of order ε^{-1} with no dependence on d and of an optimal order $\varepsilon^{-1/(1+r/d)}$ with a dependence on d. Thus, we obtain a roughly-quadratic speedup over the randomized setting and (as with the randomized algorithm) an exponential speedup over the worst-case deterministic setting.

3.2 Computational problem

We want to compute an ε-approximation of the path integral (3.3) at a given point $(\mathbf{u}, t) \in \mathbb{R}^d \times [0, \infty)$ and for arbitrary functions v, V from the class F. The definition of an ε-approximation depends on the setting; this will be made precise in the next three subsections. The ε-approximation $a_{v,V}(\mathbf{u},t)$ is computed by an algorithm A_n that uses n function values of v and V, i.e.,

$$a_{v,V}(\mathbf{u},t) = A_n\left(\mathbf{u},t,v(\mathbf{u}_1),\dots,v(\mathbf{u}_k),V(\mathbf{u}_{k+1}),\dots,V(\mathbf{u}_n)\right).$$

In the quantum setting, n denotes the number of quantum queries and classical function evaluations.

3.2.1 Worst-case deterministic setting

In the worst case setting, the error of the algorithm A_n is defined as

$$e^{\mathrm{wor}}(A_n) = \sup_{v,V\in F} |z_{v,V}(\mathbf{u},t) - a_{v,V}(\mathbf{u},t)|.$$

We also want to determine the complexity, i.e., the minimal number

$$n^{\mathrm{wor}}(\varepsilon,F) = \min\{n:\ \exists\, A_n \quad \text{such that} \quad e^{\mathrm{wor}}(A_n)\le\varepsilon\}$$

of function values that are needed to compute an ε-approximation in the worst case setting. The worst-case deterministic setting is analyzed in [26, 29].

3.2.2 Randomized setting

In this setting we use randomized algorithms and replace the worst case error assurance by an expected one. A randomized algorithm A_n depends on a random element ω chosen from some probability space Ω. More precisely, we compute

$$a_{v,V}(\mathbf{u},t;\omega) = A_{n,\omega}\left(\mathbf{u},t,v(\mathbf{u}_{\omega,1}),\dots,v(\mathbf{u}_{\omega,k}),V(\mathbf{u}_{\omega,k+1}),\dots,V(\mathbf{u}_{\omega,n_\omega})\right), \tag{3.4}$$

with $n=\mathbb{E}(n_\omega)$. This means that we allow a random choice of a mapping $A_{n,\omega}$ and sample points $\mathbf{u}_{\omega,i}$, as well as the number n_ω of sample points, whose expected value is fixed and equal to n.

We measure the randomized error of the algorithm A_n with respect to the L_2 norm, i.e.,

$$e^{\mathrm{rand}}(A_n) = \sup_{v,V\in F}\left(\mathbb{E}(z_{v,V}(\mathbf{u},t) - a_{v,V}(\mathbf{u},t;\omega))^2\right)^{1/2}.$$

To make the notation more compact we shall skip the index ω if the context clearly indicates that we deal with a randomized algorithm.

As before, we want to determine the complexity, i.e., the minimal expected number

$$n^{\mathrm{rand}}(\varepsilon, F) = \min\{n : \ \exists\, A_n \quad \text{such that} \quad e^{\mathrm{rand}}(A_n) \leq \varepsilon\}$$

of function values needed to compute an ε-approximation in the randomized setting. The randomized setting is analyzed in [27, 28].

3.2.3 Quantum setting

In the quantum setting we use quantum algorithms with randomized quantum queries; we also assume that we can perform function evaluations and arithmetic operations on a classical computer. These classical operations are used to prepare an input for a quantum algorithm and to transform the outcome of a quantum algorithm to an approximation of the exact solution. We will be interested in minimizing the total number of quantum queries and function evaluations needed to compute an ε-approximation.

We base our analysis on the simplified quantum model of computation for continuous problems with randomized queries from Section 1.1.1. We refer the reader to [4, 15, 30, 32, 52] for more detailed information. We recall that for a given class F of input functions $f : D \to C$, we want to approximate the solution operator

$$S : F \to G,$$

with G being a normed space whose norm is denoted by $\|\cdot\|_G$. A quantum algorithm may use a classical algorithm P_s with s classical function evaluations to transform a given input function $f \in F$ into $\bar{f} = P_s(f) : D \to C$, which is then used as an input to a quantum algorithm.

In this chapter we only use randomized quantum queries as defined in [52]. A quantum algorithm with randomized queries $U_{n,\omega}(\bar{f}) : \mathcal{H}_n \to \mathcal{H}_n$ is a unitary operator of the form

$$U_{n,\omega}(\bar{f}) = Q_n\, Q_{\bar{f},\omega}\, Q_{n-1} \cdots Q_1 Q_{\bar{f},\omega}\, Q_0,$$

with unitary operators $Q_0, \ldots, Q_n$ and a quantum query $Q_{\bar{f},\omega}$, for some $\bar{f} \in P_s(H)$. The query $Q_{\bar{f},\omega}$ depends on a random element $\omega \in \Omega$. This permits the computation of approximate values of $\bar{f}$ at randomized points, as explained in Section 1.1.1, see also[52].

After performing the computation, we obtain a final state

$$|\psi_{\bar{f},\omega}\rangle = U_{n,\omega}(\bar{f})|0\rangle = Q_n Q_{\bar{f},\omega} Q_{n-1} \cdots Q_1 Q_{\bar{f},\omega} Q_0 |0\rangle.$$

We then measure the final state to obtain an outcome $j \in \{0, 1, \ldots, 2^k - 1\}$ with probability

$$p_{\bar{f},\omega}(j) = |\langle \psi_{\bar{f},\omega} | j \rangle|^2.$$

Knowing the outcome j we compute the final result on a classical computer, and the quantum algorithm A_n yields

$$A_{n,\omega}(\bar{f}, j) = \phi(j).$$

for some ϕ.

The error of a quantum algorithm with randomized quantum queries A_n is defined as

$$e^{\text{quant}}(A_n) = \sup_{f \in H} \left(\mathbb{E}\, \mathbb{E}_{\text{q}}\, \| S(f) - A_{n,\omega}(P_s(f), j) \|_G^2 \right)^{1/2}, \tag{3.5}$$

where $\mathbb{E}$ is the expectation over the probability space Ω, and $\mathbb{E}_{\text{q}}$ is the expectation with respect to distribution of the quantum algorithm outcomes. We shall skip the index ω if it is clear from the context that we deal with randomized queries.

Similarly to the other settings, we want to determine the (information) complexity, i.e., the minimal number

$$n^{\text{quant}}(\varepsilon, F) = \min\{ s + n : \ \exists\, P_s\, \exists\, A_n \quad \text{such that} \quad e^{\text{quant}}(A_n, P_s) \le \varepsilon \}$$

of random quantum queries and classical function evaluations needed to guarantee that the error does not exceed ε.

Remark 3.2.1. *We briefly comment on the quantum error setting defined by (3.5). Let us concentrate for a moment on the randomness introduced by a quantum algorithm, leaving aside randomized queries. So far, the literature dealing with continuous problems in the quantum setting has mainly considered probabilistic error. That is, instead of taking an expectation with respect to all possible outcomes of a quantum algorithm, as* $\mathbb{E}_{\mathrm{q}}$ *in (3.5), we want an error estimate such that*

$$\left(\mathbb{E}\,\|S(f) - A_{n,P_s(f),\omega}\|_G^2\right)^{1/2} \le \varepsilon$$

to hold with a given (high) probability, for any $f \in F$*. Obviously these two ways of measuring the error of a quantum algorithm are related. We choose to study the average error for simplicity and two other reasons. The average error is probably more natural when we cannot verify the result of an algorithm. Moreover, (3.5) is a stronger error criterion than that above.*

The multivariate Feynman-Kac path integration problem in the quantum setting is defined by taking $f = (v, V)$, with $F \times F$ as the input function class and $S(f) = z_{v,V}(\mathbf{u}, t)$. We shall make more assumptions on the input function class in the next section.

3.3 The function class $\mathcal{B}_F$

To assure the existence of the path integral (3.3), we need to choose an appropriate class of input functions F, see also [29]:

1. To make the path integral (3.3) well defined we assume that for every $\mathbf{u} \in \mathbb{R}^d$, the functional $L_{\mathbf{u}} : F \to \mathbb{R}$ defined by $L_{\mathbf{u}} f = f(\mathbf{u})$ is continuous, and for arbitrary $a, t \in \mathbb{R}_+$ we have
$$\int_{\mathcal{C}} \|L_{\mathbf{x}(t)}\|_F \exp\left(a \int_0^t \|L_{\mathbf{x}(s)}\|_F \, ds\right) w(d\mathbf{x}) < \infty. \tag{3.6}$$
By Fernique's theorem, see e.g., [25], condition (3.6) holds if there exists $\alpha \in (0, 2)$ such that $\|L_{\mathbf{x}}\|_F = O(\|\mathbf{x}\|^\alpha)$ for $\|\mathbf{x}\|$ approaching infinity, see [39] for

details. Here and elsewhere in this chapter, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{d} x_i^2}$ denotes the Euclidean norm in $\mathbb{R}^d$.

2. We assume that F is continuously embedded into $L_\infty(\mathbb{R}^d)$. That is, $F \subset L_\infty(\mathbb{R}^d)$ and there exists a positive K such that

$$\|f\|_{L_\infty(\mathbb{R}^d)} \le K\|f\|_F \qquad \forall f \in F. \tag{3.7}$$

This assumption permits us to relate the multivariate Feynman-Kac path integration problem to uniform approximation. By uniform approximation, we mean the approximation of functions from F in the norm of $L_\infty(\mathbb{R}^d)$, i.e., given $\varepsilon > 0$ we want to find a function $\bar{f} \in L_\infty(\mathbb{R}^d)$ such that

$$\|f - \bar{f}\|_{L_\infty(\mathbb{R}^d)} \le \varepsilon\, \|f\|_F.$$

3. We assume that we can compute a uniform approximation $\bar{f}$ of the function f from the class F by a linear algorithm

$$\bar{f} = \sum_{i=1}^{n_{\mathrm{APP}}} f(\mathbf{u}_i)\,\zeta_i, \quad \mathbf{u}_i \in \mathbb{R}^d,\ \zeta_i \in L_\infty(\mathbb{R}^d) \tag{3.8}$$

that uses n_{APP} function evaluations, where

$$n_{\mathrm{APP}} = n_{\mathrm{APP}}(\varepsilon, F) = O(\varepsilon^{-\alpha(F)}) \qquad \text{as} \quad \varepsilon \to 0, \tag{3.9}$$

for some positive $\alpha(F)$. The asymptotic constant in (3.9) may depend on the dimension d. Usually the exponent $\alpha(F)$ depends on the smoothness and on the number of variables of functions from F, see Section 3.8.

We stress that these assumptions are not overly-restrictive. It is known that algorithms of the form (3.8) are optimal for the uniform approximation problem, see [33]. Moreover the number of function evaluations often depends on ε by an expression similar to (3.9), see also Section 3.8.

4. We restrict the norms of the initial value and potential functions. Namely, we assume that $\|v\|_F \leq \beta$ and $\|V\|_f \leq B$ for given positive β, B. In other words, the pair (v, V) belong to the class

$$\mathcal{B}_F = \{(f_1, f_2) \in F \times F : \|f_1\|_F \leq B, \|f_2\|_F \leq \beta\}. \tag{3.10}$$

3.4 Feynman-Kac formula as a series of multivariate integrals

In this section we briefly recall some results from [29] that are needed for our analysis.

Without loss of generality we can assume $\mathbf{u} = \mathbf{0}$ in (3.3). Then we can express the path integral as a series of multivariate integrals

$$S(v, V) := z(\mathbf{0}, t) = \sum_{k=0}^{\infty} S_k(v, V), \tag{3.11}$$

where

$$S_k(v, V) = \int_{\mathbb{R}^{(k+1)d}} v(\mathbf{z}_{k+1}) \prod_{i=1}^{k} V(\mathbf{z}_i)\, g_k(\mathbf{z}_1, \ldots, \mathbf{z}_{k+1})\, d\mathbf{z}_1 \ldots d\mathbf{z}_{k+1}, \tag{3.12}$$

with

$$g_k(\mathbf{z}_1, \ldots, \mathbf{z}_{k+1}) = \int_{0 \leq t_1 \leq \cdots \leq t_k \leq t} f_k\left(t_1, \ldots, t_k, t, \mathbf{z}_1, \ldots, \mathbf{z}_{k+1}\right) dt_1 \ldots dt_k \tag{3.13}$$

and

$$\begin{aligned} f_k\left(t_1, \ldots, t_k, t, \mathbf{z}_1, \ldots, \mathbf{z}_{k+1}\right) &= \left((2\pi)^{k+1} t_1 (t_2 - t_1) \cdots (t - t_k)\right)^{-d/2} \\ &\times \exp\left(-\frac{1}{2}\left(\frac{\|\mathbf{z}_1\|^2}{t_1} + \frac{\|\mathbf{z}_2 - \mathbf{z}_1\|^2}{t_2 - t_1} + \cdots + \frac{\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2}{t - t_k}\right)\right). \end{aligned}$$

Note that the integral (3.12) depends on the input functions v and V only through the product

$$h_k(\mathbf{z}_1, \ldots, \mathbf{z}_{k+1}) := v(\mathbf{z}_{k+1}) \prod_{i=1}^{k} V(\mathbf{z}_i). \tag{3.14}$$

Moreover, the weight functions g_k can be computed in advance, albeit with difficulty. Let us recall also that

$$\|g_k\|_{L_1(\mathbb{R}^{(k+1)d})} = \frac{t^k}{k!} \quad \text{for} \quad k \geq 0, \tag{3.15}$$

and so the norm of the weight function g_k decreases super-exponentially as k goes to infinity.

3.5 Approximating one term of the series

In this section, we present algorithms approximating one term of the series (3.11). To make the notation more clear, we define a weighted integration operator

$$I_k(f) = \int_{\mathbb{R}^{(k+1)d}} f(\mathbf{z}_1, \ldots, \mathbf{z}_{k+1})\, g_k(\mathbf{z}_1, \ldots, \mathbf{z}_{k+1})\, d\mathbf{z}_1 \ldots d\mathbf{z}_{k+1},$$

where $f : \mathbb{R}^{(k+1)d} \to \mathbb{R}$ is an integrable function. We can then rewrite one term of the series (3.11) as

$$S_k(v, V) = I_k(h_k).$$

In all three settings—worst-case deterministic, randomized and quantum—we shall use deterministic uniform approximation of the function h_k. In the worst-case setting we use uniform approximation directly as the main building block, whereas in the randomized and quantum settings, we will apply uniform approximation as a preprocessing step that will achieve variance reduction.

3.5.1 Uniform approximation by Smolyak's algorithm

Smolyak's algorithm is a powerful tool for approximating tensor product problems. For $\varphi_k \in F_k := \overbrace{F \otimes \cdots \otimes F}^{k}$, Smolyak's algorithm is of the form

$$U_{\varepsilon,k}(\varphi_k) = \sum_{i=1}^{n(\varepsilon,k)} \varphi_k(\mathbf{u}_{i,\varepsilon,1}, \ldots, \mathbf{u}_{i,\varepsilon,k})\zeta_{i,\varepsilon,k}, \tag{3.16}$$

for some $\mathbf{u}_{i,\varepsilon,j} \in \mathbb{R}^d$ and $\zeta_{i,\varepsilon,k} \in L_\infty(\mathbb{R}^{(k+1)d})$. It is proved in [29, Lemma 2] that

$$\|\varphi_k - U_{\varepsilon,k}(\varphi_k)\|_{L_\infty(\mathbb{R}^{(k+1)d})} \leq \varepsilon \|\varphi_k\|_{F_k}, \tag{3.17}$$

where

$$n(\varepsilon,k) \leq c_0 \left(c_1 + c_2 \frac{\ln 1/\varepsilon}{k-1} \right)_+^{(\alpha(F)+1)(k-1)} \varepsilon^{-\alpha(F)}, \tag{3.18}$$

for some $c_i \in \mathbb{R}$. Here a_+ denotes $\max\{a, 0\}$, and the right hand side of (3.18) is defined to be $c_0\, \varepsilon^{-\alpha(F)}$ when $k = 1$. We shall use Smolyak's algorithm to approximate the functions h_k defined by (3.14).

3.5.2 Deterministic algorithm

We first consider approximating $S_k(v,V)$ by the algorithm

$$\phi_\varepsilon^{\mathrm{det}}(v,V) = I_k(\bar{h}_{k,\varepsilon}),$$

where $\bar{h}_{k,\varepsilon} = U_{\varepsilon,k} h_k$ and h_k is defined by (3.14). Using (3.16), we can rewrite $\phi_\varepsilon^{\mathrm{det}}$ as

$$\phi_\varepsilon^{\mathrm{det}}(v,V) = \sum_{i=1}^{n(\varepsilon,k+1)} h_k(\mathbf{u}_{i,\varepsilon,1}, \ldots, \mathbf{u}_{i,\varepsilon,k+1}) I_k(\zeta_{i,\varepsilon,k+1}). \tag{3.19}$$

We stress that $I_k(\zeta_{i,\varepsilon,k+1})$ does not depend on the input functions v and V and can be precomputed.

We note that the error of the algorithm $\phi_\varepsilon^{\mathrm{det}}$ satisfies

$$|S_k(v,V) - \phi_\varepsilon^{\mathrm{det}}(v,V)| \leq \varepsilon \frac{B\beta^k t^k}{k!}. \tag{3.20}$$

Indeed by (3.10), (3.15) and (3.17) we get

$$\begin{aligned} |S_k(v,V) - \phi_\varepsilon^{\mathrm{det}}(v,V)| &\leq I_k\big(|h_k - U_{\varepsilon,k+1} h_k| \big) \\ &\leq \|h_k - U_{\varepsilon,k+1} h_k\|_{L_\infty(\mathbb{R}^{(k+1)d})}\, \|g_k\|_{L_1(\mathbb{R}^{(k+1)d})} \leq \varepsilon \frac{B\beta^k\, t^k}{k!}; \end{aligned}$$

moreover, the total number of function evaluations used by the algorithm $\phi_\varepsilon^{\mathrm{det}}(v,V)$ is $n(\varepsilon, k+1)$, as defined in (3.18).

3.5.3 Variance reduction

The idea underlying variance reduction is as follows. First we compute $\bar{h}_{k,\varepsilon} = U_{\varepsilon,k+1}(h_k)$ using $n(\varepsilon, k+1)$ function values. Then we compute

$$I_k(\bar{h}_{k,\varepsilon}) = \sum_{i=1}^{n(\varepsilon,k+1)} \bar{h}_k(\mathbf{u}_{i,\varepsilon,1}, \dots, \mathbf{u}_{i,\varepsilon,k+1}) I_k(\zeta_{i,\varepsilon,k+1}).$$

Observe that the functions $\zeta_{i,\varepsilon,k+1}$ do not depend on the input functions v and V, and so the integrals $I_k(\zeta_{i,\varepsilon,k+1})$ can be precomputed. We also stress that $\bar{h}_{k,\varepsilon}$ and $I_k(\bar{h}_{k,\varepsilon})$ are deterministic. We shall use randomized or quantum algorithms to approximate the multivariate integrals

$$I_k(h_k - \bar{h}_{k,\varepsilon}).$$

Since the error depends on the norm $\|h_k - \bar{h}_{k,\varepsilon}\|_{L_\infty(\mathbb{R}^{(k+1)d})}$, which is now small, we can do this efficiently. We present the details in the following two sections.

3.5.4 Randomized algorithm

To make our formulas shorter, we define

$$\bar{f}_{k,\varepsilon} := h_k - \bar{h}_{k,\varepsilon}.$$

We use a randomized algorithm of the form

$$\phi^{\text{rand}}_{\varepsilon,m}(v, V) = I_k(\bar{h}_{k,\varepsilon}) + Q^{\text{rand}}_m(\bar{f}_{k,\varepsilon}). \tag{3.21}$$

Here

$$Q^{\text{rand}}_m(f) = \frac{1}{m} \sum_{j=1}^{m} f(x_j) \tag{3.22}$$

denotes the classical Monte Carlo algorithm with m randomized sample points. Randomized sample points are chosen with respect to the density $g_k / \|g_k\|_{L_1(\mathbb{R}^{(k+1)d})}$.

Using the well-known error formula for the classical Monte Carlo algorithm, we conclude that

$$\begin{aligned}\left(\mathbb{E}(I_k(h_k) - \phi^{\mathrm{rand}}_{\varepsilon,m}(v,V))^2\right)^{1/2} &= \left(\mathbb{E}\left(I_k(\bar{f}_{k,\varepsilon}) - Q^{\mathrm{rand}}_m(\bar{f}_{k,\varepsilon})\right)^2\right)^{1/2} \\ &= \frac{1}{\sqrt{m}}\left(\mathrm{Var}(\bar{f}_{k,\varepsilon})\right)^{1/2}, \quad (3.23)\end{aligned}$$

with

$$\mathrm{Var}(\bar{f}_{k,\varepsilon}) = I_k(\bar{f}^{\,2}_{k,\varepsilon}) - \left(I_k(\bar{f}_{k,\varepsilon})\right)^2.$$

Clearly, from (3.17) and then from (3.10) and (3.15), we get

$$\left(\mathrm{Var}(\bar{f}_{k,\varepsilon})\right)^{1/2} \le \frac{t^k}{k!}\,\|\bar{f}_{k,\varepsilon}\|_{L_\infty(\mathbb{R}^{(k+1)d})} \le \varepsilon\,\frac{\|v\|_F\,\|V\|_F^k\,t^k}{k!} \le \varepsilon\,\frac{B\beta^k\,t^k}{k!}. \tag{3.24}$$

This yields the error estimate

$$\left(\mathbb{E}(I_k(h_k) - \phi^{\mathrm{rand}}_{\varepsilon,m}(v,V))^2\right)^{1/2} \le \frac{\varepsilon}{\sqrt{m}}\,\frac{B\beta^k\,t^k}{k!}, \tag{3.25}$$

and the total number of function evaluations being

$$n(\varepsilon, k+1) + m. \tag{3.26}$$

3.5.5 Quantum algorithm

The structure of the quantum algorithm is similar to the randomized one and has the form

$$\phi^{\mathrm{quant}}_{\varepsilon,m,\kappa}(v,V) = I_k(\bar{h}_{k,\varepsilon}) + Q^{\mathrm{quant}}_{m,\kappa}(\bar{f}_{k,\varepsilon}), \tag{3.27}$$

with, as before, $\bar{f}_{k,\varepsilon} = h_k - \bar{h}_{k,\varepsilon}$. Here, we use a quantum algorithm $Q^{\mathrm{quant}}_{m,\kappa}$, with κ randomized quantum queries, that approximates the classical Monte Carlo algorithm (3.22). In [19], the problem of approximating

$$\frac{1}{m}\sum_{j=1}^{m} f(x_j)$$

was analyzed for Boolean functions f. By reducing the summation problem for bounded real functions to the summation problem for Boolean functions as in [15], we see that a result similar to that of [19] holds. From [19] and (3.17) we conclude that

$$\left(\mathbb{E}_{\mathrm{q}}\left(\frac{1}{m}\sum_{j=1}^{m} f(x_j) - Q_{m,\kappa}^{\mathrm{quant}}(\bar{f}_{k,\varepsilon})\right)^2\right)^{1/2} = O\left(\frac{1}{\kappa}\|\bar{f}_{k,\varepsilon}\|_{L_\infty(\mathbb{R}^{(k+1)d})}\right) = O\left(\frac{\varepsilon}{\kappa}\,B\beta^k\right).$$

By integrating over randomized sample points, we obtain

$$\left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}\left|Q_{m,k}^{\mathrm{rand}}(\bar{f}_{k,\varepsilon}) - Q_{m,\kappa}^{\mathrm{quant}}(\bar{f}_{k,\varepsilon})\right|^2\right)^{1/2} = O\left(\frac{\varepsilon}{\kappa}\,\frac{B\beta^k\,t^k}{k!}\right). \tag{3.28}$$

The total number of randomized quantum queries and function evaluations is

$$n(\varepsilon, k+1) + \kappa.$$

We stress that this number does not depend on m, which is only used for the definition of the Monte Carlo algorithm providing sample points for the quantum algorithm.

We now estimate the total error as

$$\begin{aligned}\left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}(I_k(h_k) - \phi_{\varepsilon,m,\kappa}^{\mathrm{quant}}(v,V))^2\right)^{1/2} \le{}& \left(\mathbb{E}\left(I_k(\bar{f}_{k,\varepsilon}) - Q_m^{\mathrm{rand}}(\bar{f}_{k,\varepsilon})\right)^2\right)^{1/2}\\ &+ \left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}\left|Q_m^{\mathrm{rand}}(\bar{f}_{k,\varepsilon}) - Q_{m,\kappa}^{\mathrm{quant}}(\bar{f}_{k,\varepsilon})\right|^2\right)^{1/2}.\end{aligned}$$

This, by (3.25) and (3.28), yields

$$\left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}(I_k(h_k) - \phi_{\varepsilon,m,\kappa}^{\mathrm{quant}}(v,V))^2\right)^{1/2} = O\left(\frac{\varepsilon}{\sqrt{m}}\,\frac{B\beta^k\,t^k}{k!} + \frac{\varepsilon}{\kappa}\,\frac{B\beta^k\,t^k}{k!}\right).$$

Letting $m = \kappa^2$ we get the error bound

$$\left(\mathbb{E}\,\mathbb{E}_{\mathrm{q}}(I_k(h_k) - \phi_{\varepsilon,m,\kappa}^{\mathrm{quant}}(v,V))^2\right)^{1/2} = O\left(\frac{\varepsilon}{\kappa}\,\frac{2B\,\beta^k\,t^k}{k!}\right) \tag{3.29}$$

using

$$n(\varepsilon, k+1) + \kappa \tag{3.30}$$

function values and quantum queries. For the sake of convenience we denote

$$\phi_{\varepsilon,\kappa}^{\mathrm{quant}} = \phi_{\varepsilon,m,\kappa}^{\mathrm{quant}} \qquad \text{with } m = \kappa^2.$$

3.6 Complete algorithms

Based on the previous two sections, we are ready to present algorithms computing an ε-approximation of multivariate Feynman-Kac path integral $S(v, V)$. We simply approximate consecutive terms of the series

$$S(v, V) = \sum_{k=0}^{\infty} S_k(v, V)$$

by the algorithms

$$\phi^{\mathrm{det}}_{\varepsilon^{\mathrm{det}}_k} \qquad \text{or} \qquad \phi^{\mathrm{rand}}_{\varepsilon^{\mathrm{rand}}_k, m_k} \qquad \text{or} \qquad \phi^{\mathrm{quant}}_{\varepsilon^{\mathrm{quant}}_k, \kappa_k}.$$

Here, the accuracies $\varepsilon^{\mathrm{det}}_k$, $\varepsilon^{\mathrm{rand}}_k$ and $\varepsilon^{\mathrm{quant}}_k$ in the corresponding settings are

$$\varepsilon^{\mathrm{det}}_k = \varepsilon \, \frac{k!}{B\beta^k \, t^k 2^{k+1}}, \tag{3.31}$$

$$\varepsilon^{\mathrm{rand}}_k = \varepsilon^{2/(\alpha(F)+2)} \frac{k!}{B\beta^k \, t^k 2^{k+1}}, \tag{3.32}$$

$$\varepsilon^{\mathrm{quant}}_k = \varepsilon^{1/(\alpha(F)+1)} \frac{k!}{B\beta^k \, t^k 2^{k+2}}, \tag{3.33}$$

with the numbers of randomized sample points m_k and quantum queries κ_k being

$$m_k = \left\lceil \varepsilon^{-2\alpha(F)/(\alpha(F)+2)} \right\rceil, \quad \kappa_k = \left\lceil \varepsilon^{-\alpha(F)/(\alpha(F)+1)} \right\rceil. \tag{3.34}$$

Since m_k and κ_k are independent of k, we shall drop the indices and write $m = m_k$ and $\kappa = \kappa_k$.

Remark 3.6.1. *We see from the definitions (3.31), (3.32) and (3.33) that for k going to infinity, we have* $\varepsilon^{\mathrm{det}}_k$, $\varepsilon^{\mathrm{rand}}_k$ *and* $\varepsilon^{\mathrm{quant}}_k$ *also tending to infinity super-exponentially. Then, by (3.7) and then by (3.20), (3.25) and (3.29), we see that for* $k = O(\ln \varepsilon^{-1})$ *the deterministic zero algorithm provides sufficient accuracy in all three settings. See also the following three subsections.*

We are now ready to present the final forms of the algorithms and analyze their errors and costs.

3.6.1 Deterministic algorithm

The deterministic algorithm is of the form

$$\Phi_\varepsilon^{\text{det}}(v,V)=\sum_{k=0}^{N_\varepsilon^{\text{det}}}\phi^{\text{det}}_{\varepsilon_k^{\text{det}}}(v,V),$$

with the finite sum limit $N_\varepsilon^{\text{det}}$. From Remark 3.6.1 we see that for $\varepsilon_k^{\text{det}}\geq K^{k+1}$, the zero algorithm yields a sufficient accuracy so we can define

$$N_\varepsilon^{\text{det}}=\min\{k\in\mathbb{N}:\varepsilon_k^{\text{det}}\geq K^{k+1}\}$$

and $N_\varepsilon^{\text{det}}=O(\ln\varepsilon^{-1})$. The error of the algorithm $\Phi_\varepsilon^{\text{det}}$ can be estimated from (3.20) and (3.31) by

$$|S(v,V)-\Phi_\varepsilon^{\text{det}}(v,V)|\leq\sum_{k=1}^{\infty}|I_k(h_k)-\phi^{\text{det}}_{\varepsilon_k^{\text{det}}}(v,V)|\leq\varepsilon. \tag{3.35}$$

The number $n(\Phi_\varepsilon^{\text{det}})$ of function evaluations used by $\Phi_\varepsilon^{\text{det}}$ satisfies

$$n(\Phi_\varepsilon)=O(\varepsilon^{-\alpha(F)-\delta})\qquad\forall\delta>0. \tag{3.36}$$

The proof is given in [29] and is similar to the one from [39]. First, we have

$$n(\Phi_\varepsilon^{\text{det}})\leq\sum_{k=0}^{\infty}n(\varepsilon_k,k+1).$$

Using (3.18) and (3.31), we obtain the bound

$$n(\Phi_\varepsilon^{\text{det}})=O\left(\left(1+\sum_{k=1}^{\infty}\left(c_1+c_2\frac{\ln 1/\varepsilon_k^{\text{det}}}{k}\right)_+^{(\alpha(F)+1)k}\left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)}\right)\varepsilon^{-\alpha(F)}\right)$$

and it can be shown, similarly to [39], that

$$\sum_{k=1}^{\infty}\left(c_1+c_2\frac{\ln 1/\varepsilon_k^{\text{det}}}{k}\right)_+^{(\alpha(F)+1)k}\left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)}=O(\varepsilon^{-\delta})$$

for all $\delta>0$.

3.6.2 Randomized algorithm

The randomized algorithm approximating $S(v,V)$ is of the form

$$\Phi_\varepsilon^{\mathrm{rand}}(v,V)=\sum_{k=0}^{N_\varepsilon^{\mathrm{rand}}}\phi^{\mathrm{rand}}_{\varepsilon_k^{\mathrm{rand}},m},$$

with $N_\varepsilon^{\mathrm{rand}}$ defined as

$$N_\varepsilon^{\mathrm{rand}}=\min\left\{k\in\mathbb{N}:\varepsilon_k^{\mathrm{rand}}\geq m\,K^{k+1}\right\}.$$

Again $N_\varepsilon^{\mathrm{rand}}=O(\ln\varepsilon^{-1})$. As in Section 3.6.1, by (3.25) and (3.32), we can prove that the error of the algorithm $\Phi_\varepsilon^{\mathrm{rand}}$ satisfies

$$\left(\mathbb{E}(S(v,V)-\Phi_\varepsilon^{\mathrm{rand}}(v,V))^2\right)^{1/2}\leq\sum_{k=1}^{\infty}\left(\mathbb{E}(I_k(h_k)-\phi^{\mathrm{rand}}_{\varepsilon_k^{\mathrm{rand}},m}(v,V))^2\right)^{1/2}\leq\varepsilon \qquad (3.37)$$

The number $n(\Phi_\varepsilon^{\mathrm{rand}})$ of function evaluations used by the algorithm $\Phi_\varepsilon^{\mathrm{rand}}$ satisfies

$$n\left(\Phi_\varepsilon^{\mathrm{rand}}\right)=O\left(\varepsilon^{-2\alpha(F)/(\alpha(F)+2)-\delta}\right)\qquad\forall\delta>0. \qquad (3.38)$$

The proof is based on an argument similar to that from Section 3.6.1. By the bounds (3.18), (3.26) and (3.34) we can estimate $n(\Phi_\varepsilon^{\mathrm{rand}})$ by

$$n\left(\Phi_\varepsilon^{\mathrm{rand}}\right)=O\left(\left(1+B^{\alpha(F)}+\sum_{k=1}^{\infty}\left(c_1+c_2\frac{\ln 1/\varepsilon_k^{\mathrm{rand}}}{k}\right)_+^{(\alpha(F)+1)k}\times\left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)}\right)\varepsilon^{-2\alpha(F)/(\alpha(F)+2)}\right).$$

Similarly to Section 3.6.1, we can show that

$$\sum_{k=1}^{\infty}\left(c_1+c_2\frac{\ln 1/\varepsilon_k^{\mathrm{rand}}}{k}\right)_+^{(\alpha(F)+1)k}\left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)}=O(\varepsilon^{-\delta}),$$

for all $\delta>0$, which proves (3.38).

3.6.3 Quantum algorithm

The quantum algorithm $\Phi_\varepsilon^{\text{quant}}$ is defined as

$$\Phi_\varepsilon^{\text{quant}}(v,V) = \sum_{k=0}^{N_\varepsilon^{\text{quant}}} \phi^{\text{quant}}_{\varepsilon_k^{\text{quant}},\kappa},$$

with $N_\varepsilon^{\text{quant}}$ satisfying

$$N_\varepsilon^{\text{quant}} = \min\left\{k \in \mathbb{N} : \varepsilon_k^{\text{rand}} \geq \kappa\, K^{k+1}\right\},$$

and $N_\varepsilon^{\text{quant}} = O(\ln \varepsilon^{-1})$. As before, we can easily prove that the error of the algorithm is of order ε. Indeed, by (3.29) and (3.33) we get

$$\left(\mathbb{E}\,\mathbb{E}_{\text{q}}(S(v,V) - \Phi_\varepsilon^{\text{quant}}(v,V))^2\right)^{1/2} \leq \sum_{k=0}^{\infty} \left(\mathbb{E}\,\mathbb{E}_{\text{q}}(I_k(h_k) - \phi^{\text{quant}}_{\varepsilon_k^{\text{quant}},\kappa}(v,V))^2\right)^{1/2} = O(\varepsilon).$$

The number of function evaluations and quantum queries $n(\Phi_\varepsilon^{\text{quant}})$ of the algorithm $\Phi_\varepsilon^{\text{quant}}$ can be estimated as

$$n\left(\Phi_\varepsilon^{\text{quant}}\right) = O\left(\varepsilon^{-\alpha(F)/(\alpha(F)+1)-\delta}\right) \qquad \forall \delta > 0. \tag{3.39}$$

As in Section 3.6.2 the proof follows from the estimate

$$n\left(\Phi_\varepsilon^{\text{quant}}\right) = O\left(\left(B^{\alpha(F)} + \sum_{k=1}^{\infty}\left(c_1 + c_2\frac{\ln 1/\varepsilon_k^{\text{quant}}}{k}\right)_+^{(\alpha(F)+1)k} \times \left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)} + 1\right)\varepsilon^{-\alpha(F)/(\alpha(F)+1)}\right)$$

and the fact that

$$\sum_{k=1}^{\infty}\left(c_1 + c_2\frac{\ln 1/\varepsilon_k^{\text{quant}}}{k}\right)_+^{(\alpha(F)+1)k} \left(\frac{B\beta^k t^k 2^{k+1}}{k!}\right)^{\alpha(F)} = O(\varepsilon^{-\delta})$$

for all $\delta > 0$.

Obviously we can obtain the error estimate

$$\left(\mathbb{E}\,\mathbb{E}_{\text{q}}(S(v,V) - \Phi_\varepsilon^{\text{quant}}(v,V))^2\right)^{1/2} \leq \varepsilon \tag{3.40}$$

by redefining ε modulo a factor that would increase the asymptotic constant in the estimate of the number of function evaluations and quantum queries (3.38).

3.7 Complexity of multivariate Feynman-Kac path integration

The analysis of the complexity of the multivariate Feynman-Kac path integration in randomized and quantum settings is based on that presented in [29] and [39].

3.7.1 Lower bounds

Lower bounds for our problem complexities are provided by the complexities of multivariate weighted integration problem. By this problem, we mean an approximation of the integration operator $I : F \to \mathbb{R}$ defined by

$$I(f) = (2\pi t)^{-d/2} \int_{\mathbb{R}^d} f(\mathbf{u}) \exp(-\|\mathbf{u}\|/(2t))\, d\mathbf{u} \qquad \forall\, f \in \mathcal{B}_F.$$

1. Let A_n^{det} be a deterministic algorithm that uses n function values and approximates the integration operator I. We say that this algorithm computes an ε-approximation of the weighted integral if

   $$|I(f) - A_n^{\text{det}}(f)| \le \varepsilon \qquad \forall\, f \in F.$$

   We denote by $n_{\text{INT}}^{\text{det}}(\varepsilon, \mathcal{B}_F)$ the minimal number of function values needed to compute an ε-approximation in the worst-case deterministic setting for the class $\mathcal{B}_F$.

2. Let A_n^{rand} be a randomized algorithm that uses n function values and approximates the integration operator I. We say that this algorithm computes an ε-approximation of the weighted integral if

   $$\left(\mathbb{E}\left(I(f) - A_n^{\text{rand}}(f)\right)^2\right)^{1/2} \le \varepsilon \qquad \forall\, f \in \mathcal{B}_F.$$

   We denote by $n_{\text{INT}}^{\text{rand}}(\varepsilon, \mathcal{B}_F)$ the minimal number of function values needed to compute an ε-approximation in the randomized setting.

3. Let A_n^{quant} be a quantum algorithm that uses n randomized quantum queries and approximates the operator I. We say that A_n^{quant} computes an ε-approximation of the weighted integral if
$$\left(\mathbb{E}\,\mathbb{E}_{\text{q}}\left(I(f) - A_n^{\text{quant}}(f)\right)^2\right)^{1/2} \le \varepsilon \qquad \forall\, f \in \mathcal{B}_F. \tag{3.41}$$
We define $n_{\text{INT}}^{\text{quant}}(\varepsilon, \mathcal{B}_F)$ as the minimal number of quantum queries needed to compute an ε-approximation.

Since $S(v,0) = I(v)$, as in [29], we can reduce multivariate Feynman-Kac path integration to multivariate integration with Gaussian weight by taking $V \equiv 0$. Moreover, (3.10) and (3.7) imply that
$$\begin{aligned} n_{\text{INT}}^{\text{det}}(\varepsilon, \mathcal{B}_F) &\le n^{\text{det}}(\varepsilon, \mathcal{B}_F), \\ n_{\text{INT}}^{\text{rand}}(\varepsilon, \mathcal{B}_F) &\le n^{\text{rand}}(\varepsilon, \mathcal{B}_F), \\ n_{\text{INT}}^{\text{quant}}(\varepsilon, \mathcal{B}_F) &\le n^{\text{quant}}(\varepsilon, \mathcal{B}_F). \end{aligned}$$

3.7.2 Upper bounds

Obvious estimates of the complexity of the multivariate Feynman-Kac path integration are provided by the (information) cost of the algorithms derived in Section 3.6. Thus, by (3.36), (3.38) and (3.39) we get
$$\begin{aligned} n^{\text{det}}(\varepsilon, \mathcal{B}_F) &= O\left(\varepsilon^{-\alpha(F))-\delta}\right), \\ n^{\text{rand}}(\varepsilon, \mathcal{B}_F) &= O\left(\varepsilon^{-2\alpha(F)/(\alpha(F)+2)-\delta}\right), \\ n^{\text{quant}}(\varepsilon, \mathcal{B}_F) &= O\left(\varepsilon^{-\alpha(F)/(\alpha(F)+1)-\delta}\right) \end{aligned}$$
for all $\delta > 0$, where $\alpha(F)$ is the exponent of the uniform approximation problem complexity for the space F containing the class $\mathcal{B}_F$ defined by (3.9).

From the previous two sections, we can determine when the deterministic, randomized and quantum algorithms presented here are almost optimal. This happens for the classes of input functions for which worst-case deterministic, randomized and quantum complexities of the integration problem defined in Section 3.7.1 are of orders $\varepsilon^{-\alpha(F)}$, $\varepsilon^{-2\alpha(F)/(\alpha(F)+2)}$ and $\varepsilon^{-\alpha(F)/(\alpha(F)+1)}$ respectively. We present two examples of such classes in the next section.

3.8 Examples

In this section we present two examples of function classes F satisfying the assumptions from Section 3.3 and compute lower and upper bounds of of the complexities of the multivariate Feynman-Kac path integration. For both examples, the algorithms presented in this chapter are almost optimal.

3.8.1 The Sobolev space of compactly supported functions

Let F be a class of d variate r times continuously differentiable functions whose supports are contained in a cube $[a,b]^d \subset \mathbb{R}^d$. Thus F is a subclass of the Sobolev space $W_\infty^{r,d}([a,b]^d)$ with the norm

$$\|f\|_{W_\infty^{r,d}([a,b]^d)} = \sum_{|\boldsymbol{\alpha}|\le r} \|f^{(\boldsymbol{\alpha})}\|_{L_\infty([a,b]^d)},$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_d] \in \mathbb{N}^d$ and $f^{(\boldsymbol{\alpha})} = \partial^{|\boldsymbol{\alpha}|}/\partial^{\alpha_1} \cdots \partial^{\alpha_d}$.

Clearly, assumptions 1 and 2 of Section 3.3 are satisfied. From [6, 7, 33], we also know an optimal uniform approximation algorithm that satisfies assumption 3 with the exponent $\alpha(F) = d/r$. Based on this algorithm, we can construct almost optimal algorithms $\Phi_\varepsilon^{\mathrm{det}}$, $\Phi_\varepsilon^{\mathrm{rand}}$ and $\Phi_\varepsilon^{\mathrm{quant}}$ computing ε-approximations of the multivariate Feynman-Kac path integral (3.3) for $(v, V) \in \mathcal{B}_F$ in the sense of (3.35), (3.37) and (3.40), with the number of function evaluations and/or quantum queries being roughly of order $\varepsilon^{-d/r}$, $\varepsilon^{-2/(1+2r/d)}$ and $\varepsilon^{-1/(1+r/d)}$. We see that the cost of the deterministic algorithm depends exponentially on the dimension d, whereas for the randomized and quantum algorithms the exponents of ε^{-1} are at most 2 and 1 respectively. Thus, the curse of dimensionality present in the worst-case deterministic setting is vanquished in both the randomized and quantum settings. This corresponds to the case when the exponent of ε^{-1} is 2 in the randomized setting and 1 in the quantum setting. We can decrease these exponents to $2/(1+2r/d)$ and $1/(1+r/d)$, at the expense of introducing a dependence on the number d of variables.

3.8.2 Periodic functions

Although this example was considered in [29], we repeat the details for the reader's convenience. Following [45] we consider the class Ψ of 2π-periodic functions $f : [0,2\pi]^d \to \mathbb{R}^d$ satisfying the condition

$$\forall\, f \in \Psi \quad \forall j = 1, \ldots, d \quad \exists\, \varphi_j \in L_\infty([-2\pi, 2\pi]^d)$$

$$f(\mathbf{x}) = \frac{1}{2\pi} \int_0^{2\pi} \varphi_j(x_1, \ldots, x_j - t, \ldots, x_d)\, F_r(t)\, dt, \quad (3.42)$$

where $r > 0$ and

$$F_r(t) = 1 + 2 \sum_{k=0}^{\infty} k^{-r} \cos\left(k\,t - \frac{r\,\pi}{2}\right).$$

The norm in the class Ψ is defined as

$$\|f\|_\Psi = \frac{1}{d} \sum_{j=1}^{d} \|\varphi_j\|_{L_\infty([-2\pi,2\pi]^d)},$$

where the φ_j are functions from the representation (3.42) of the function f. The paper [45] provides a linear algorithm U_ε that computes a uniform ε-approximation of functions from the class Ψ, so that

$$\|f - U_\varepsilon f\|_{L^\infty([0,2\pi]^d)} \le \varepsilon\, \|f\|_\Psi \qquad \forall f \in \Psi,$$

with a cost of order $\varepsilon^{-d/r}$.

Let F denote the class of functions $f : \mathbb{R}^d \to \mathbb{R}$ that are periodic extensions of functions from Ψ. Let $\|f\|_F := \|f|_{[0,2\pi]^d}\|_\Psi$. The problem of *uniform* approximation for the class F can be obviously solved using the algorithm mentioned above, with the same cost as for the class Ψ. Similarly to the previous example, we have to check the assumptions of Section 3.3. It is easy to see that for $f \in F$, $\mathbf{z} \in \mathbb{R}^d$, and arbitrary $j \in \{1, 2, \ldots, d\}$, we have

$$f(\mathbf{z}) \leq \|f\|_{L^\infty(\mathbb{R}^d)} = \|f|_{[0,2\pi]^d}\|_{L^\infty([0,2\pi]^d)} \leq C\,\|\varphi_j\|_{L^\infty([-2\pi,2\pi]^d)},$$

with $C = (2\pi)^{-1}\int_0^{2\pi} |F_r(t)|\,dt$. Hence

$$|f(\mathbf{z})| \leq \|f\|_{L^\infty(\mathbb{R}^d)} \leq C\,\|f\|_F$$

and so function evaluations are continuous. The remaining assumptions follow immediately.

Finally we can construct the algorithms $\Phi_\varepsilon^{\mathrm{det}}$, $\Phi_\varepsilon^{\mathrm{rand}}$ and $\Phi_\varepsilon^{\mathrm{quant}}$ as in the previous example. Since the uniform approximation exponent $\alpha(F)$ is the same for both examples the algorithms approximating multivariate Feynman-Kac path integrals for the space of periodic functions have properties similar to the algorithms from the previous example.

3.9 Conclusions

The results of this chapter provide estimates of the complexity of multivariate Feynman-Kac path integration in the worst-case deterministic, randomized and quantum settings. We also present optimal algorithms in these three settings. Although this problem considerably differs from ordinary multivariate integration, the complexity estimates that we obtain are similar to those for multivariate integration.

In Section 1.2.2.2, we indicated that one of our goals is to compare the complexity of multivariate Feynman-Kac path integration to the complexity of general path integration considered in [48]. This paper deals with path integrals over a separable Banach space X, with respect to a zero mean Gaussian measure μ, with eigenvalues $\lambda_i = \Theta(i^{-k})$, $k > 1$. For a functional $f : X \to \mathbb{R}$ being s times Frechet differentiable, the general path integral is defined as

$$S(f) = \int_X f(x)\,\mu(dx). \tag{3.43}$$

We stress that in [48] evaluations of the functional f from (3.43) are used, while for multivariate Feynman-Kac path integration we do not assume that we can compute the values of the integrand in (3.3). Instead we can only compute the values of the input functions—initial condition and potential. We compare the results from [48] with the complexities of multivariate Feynman-Kac path integration for the input function class considered in Section 3.8.1.

The cost of the worst-case deterministic algorithm of [48] is of order $\varepsilon^{-\varepsilon^{-\gamma(s)/k}}$, with $\gamma(s) = 1 + \delta_{s,1}$. This estimate is significantly larger than the complexity order $\varepsilon^{-d/r}$ for multivariate Feynman-Kac path integration. Since the deterministic algorithm presented in this chapter is optimal, its cost order is also $\varepsilon^{-d/r}$ and we see that in the deterministic setting the special structure of the Feynman-Kac path integral provides a significant improvement.

The randomized algorithm for general path integration of [48] has the cost order $\varepsilon^{-2-\gamma(s)/k}$, whereas the complexity of the multivariate Feynman-Kac path integration (with input functions as in Section 3.8) in the randomized setting is of order $\varepsilon^{-2/(1+2r/d)}$. This is also the order of the optimal randomized algorithm presented in this chapter. Thus, our algorithm has a polynomial gain for moderate r/d. The same can be said about the quantum algorithm of [48], whose cost order is ε^{-1}, while our algorithm has the cost of order $\varepsilon^{-1/(1+r/d)}$. Thus for moderate d/r the special structure of the Feynman-Kac path integral yields an improvement in both the randomized and quantum settings.

Chapter 4

Open problems

The analysis of Boolean summation and multivariate Feynman-Kac path integration from Chapters 2 and 3 leaves some open problems and possible extensions. In this chapter we present three research problems that seem to be interesting and challenging.

4.1 Average-average error for quantum Boolean summation

The quantum Boolean summation was analyzed in Chapter 2. We recall that the performance of the **QS** algorithm is measured with respect to the input Boolean function class $\mathbb{B}_N$ and the final state distribution. This corresponds to the four settings—worst-probabilistic, average-probabilistic, worst-average and average-average. We studied the worst performance with respect to the class $\mathbb{B}_N$ and the probabilistic and average performance with respect to the final state distribution. We also studied the average performance with respect to the class $\mathbb{B}_N$ and the probabilistic performance with respect to the final state distribution.

What remains to be studied is the average-average error, which measures the average performance with respect to both a distribution $\mathbf{p}$ on the Boolean function class $\mathbb{B}_N$ and the distribution of the $\mathbf{QS}$ algorithm final state. We define the average-average error as follows

$$e_q^{\text{avg-avg}}(\mathbf{QS}) = \left(\sum_{f \in \mathbb{B}_N} \sum_{j=0}^{M-1} \mathbf{p}(f) p_f(j) \left| S(f) - \mathbf{QS}(f,j) \right|^q \right)^{1/q} .$$

As with the worst-average error we measure the average performance in the L_q norm, we allow $q \in [1, \infty]$. Similarly to the average-probabilistic error we may consider two measures on the set $\mathbb{B}_N$: $\mathbf{p}_1$—the uniform distribution over Boolean functions, and $\mathbf{p}_2$—the uniform distribution over the means. We stress that the average-average error setting for the measure $\mathbf{p}_2$ and $q = 1$ provides a lower bound for the complexity of quantum summation and integration when randomized queries are used. Lower bounds for the general quantum Boolean summation problem were established in [36]. We expect that for uniformly distributed Boolean functions (the measure $\mathbf{p}_1$), we shall observe that the cost of the $\mathbf{QS}$ algorithm will on M, as in the worst-probabilistic setting. For the uniformly distributed means (the measure $\mathbf{p}_2$) the results are probably quite different.

4.2 Multivariate Feynman-Kac path integration in the quantum setting with deterministic queries

We used the concept of randomized query developed in [52] to study the multivariate Feynman-Kac path integration in the quantum setting in Chapter 3. It would be natural to consider deterministic (bit) queries used in studies of almost all continuous problems mentioned in Section 1.1. However, as we mentioned before, the analysis of the multivariate Feynman-Kac path integration in the quantum setting leads to some technical difficulties. That is why our results has been so far established only for randomized queries. It would be important to determine whether the power of deterministic queries is comparable to that of randomized queries.

4.3 Multivariate Feynman-Kac path integration in finite-order weighted spaces

The notion of finite-order weighted spaces, see e.g., [44, 49, 50], has recently been studied for multivariate problems. Finite-order weights are used to model continuous multivariate problems for which d-variate functions can be decomposed as sums of functions of fewer variables. The weights describe the relative importance of each group of variables. Multivariate problems over finite-order weighted spaces are often tractable. That is, the minimal number of function values needed to compute an ε-approximation is polynomial in ε^{-1} and d. The finite-order weighted structure of a function space seems to also be very promising for multivariate Feynman-Kac path integration. The integrands that are used in applications often have this kind of structure. Moreover, uniform approximation, which is used in optimal algorithms for all three settings discussed in Chapter 3, is tractable for finite-order weighted spaces, see [49]. Therefore, it would be important to also study multivariate Feynman-Kac for finite-order weighted spaces in the quantum model of computation.

Bibliography

[1] D. S. Abrams and S. Lloyd, *Quantum algorithm providing exponential speedup increase for finding eigenvalues and eigenvectors*, Phys. Rev. Lett. **83** (1999), 5162–5165.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, *Quantum lower bounds by polynomials*, Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FoCS '98), 1998, pp. 352–361.

[3] A. J. Bessen, *Lower bound for quantum phase estimation*, Phys. Rev. A **71** (2005), 042313.

[4] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum amplitude estimation and amplification*, Contemporary Mathematics **305** (2002), 53–74.

[5] A. J. Chorin, *Accurate evaluation of Wiener integrals*, Math. Comput. **27** (1973), 1–15.

[6] W. Dahmen, R. DeVore, and K. Scherer, *Multi-dimensional spline approximation*, SIAM J. Numer. Anal. **17** (1980), 380–402.

[7] C. de Boor and G. Fix, *Spline approximation by quasi-interpolants*, J. Approximation Theory **8** (1973), 19–45.

[8] D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. R. Soc. Lond., Ser. A 400, 1985, pp. 97–117.

[9] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proc. R. Soc. Lond., Ser. A 439, 1992, pp. 553–.

[10] J. W. Eaton, *GNU Octave Manual*, Network Theory, 2002.

[11] A. D. Egorov, P. I. Sobolevsky, and L. A. Yanovich, *Functional Integrals: Approximate Evaluations and Applications*, Kluwer Acad. Publ., 1993.

[12] R. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21** (1982), 467–488.

[13] L. Grover, *A fast quantum mechanical algorithm for database search*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC), 1996, pp. 212–219.

[14] O. H. Hald, *Approximation of Wiener integrals*, J. Comput. Phys. **69** (1987), 460–470.

[15] S. Heinrich, *Quantum summation with an application to integration*, J. Complexity **18** (2002), 1–50.

[16] ______, *Quantum integration in Sobolev classes*, J. Complexity **19** (2003), 19–42.

[17] ______, *Quantum approximation I. Embeddings of finite dimensional L_p spaces*, J. Complexity **20** (2004), 5–26.

[18] ______, *Quantum approximation II. Sobolev embeddings*, J. Complexity **20** (2004), 27–45.

[19] S. Heinrich, M. Kwas, and H. Woźniakowski, *Quantum Boolean summation with repetitions in the worst-average setting*, Proceedings of the 5th International Conference on Monte Carlo and Quasi-Monte Carlo Methods, Springer-Verlag, 2003.

[20] S. Heinrich and E. Novak, *On a problem in quantum summation*, J. Complexity **19** (2001), 1–18.

[21] ———, *Optimal summation and integration by deterministic, randomized and quantum algorithms*, Proceedings of the 4th International Conference on Monte Carlo and Quasi Monte Carlo Methods 2000 (K.-T. Fang, F. J. Hickernell, and H. Niederreiter, eds.), Springer-Verlag, 2002, pp. 50–62.

[22] P. Jaksch and A. Papageorgiou, *Eigenvector approximation leading to exponential quantum speedup of quantum eigenvalue calculation*, Phys. Rev. Lett. **91** (2003), no. 25.

[23] Z. Kacewicz, *Randomized and quantum algorithms yield a speed-up for initial-value problems*, J. Complexity **20** (2004), 821–834.

[24] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer-Verlag, 1988.

[25] H.-H. Kuo, *Gaussian Measures in Banach Spaces*, Springer-Verlag, 1995.

[26] M. Kwas, *Worst case complexity of multivariate Feynman-Kac path integration*, Master's thesis, University of Warsaw, 2001, in Polish.

[27] ———, *Complexity of multivariate Feynman-Kac path integration in randomized and quantum settings*, in preparation, 2004.

[28] ———, *An optimal Monte Carlo algorithm for multivariate Feynman-Kac path integrals*, submitted, 2004.

[29] M. Kwas and Y. Li, *Worst case complexity of multivariate Feynman-Kac path integration*, J. Complexity **19** (2003), 730–743.

[30] M. Kwas and H. Woźniakowski, *Sharp error bounds on quantum Boolean summation in various settings*, J. Complexity **20** (2004), 669–698.

[31] A. Nayak and F. Wu, *The quantum query complexity of approximating the median and related statistics*, Proceedings of the 31th Annual ACM Symposium on the Theory of Computing (STOC), 1999, pp. 384–393.

[32] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2001.

[33] E. Novak, *Deterministic and Stochastic Error Bounds in Numerical Analysis*, Lecture Notes in Mathematics, vol. 1349, Springer, 1988.

[34] ______, *Quantum complexity of integration*, J. Complexity **17** (2001), 2–16.

[35] E. Novak, I. H. Sloan, and H. Woźniakowski, *Tractability of approximation for weighted Korobov spaces on classical and quantum computers*, Foundation of Computatonal Mathematics **4** (2004), 121–156.

[36] A. Papageorgiou, *Average case quantum lower bounds for computing the Boolean means*, J. Complexity **20** (2004), 713–731.

[37] A. Papageorgiou and H. Woźniakowski, *Classical and quantum complexity of the Sturm-Liouville eigenvalue problem.*

[38] ______, *The Sturm-Liouville eigenvalue problem and NP-complete problems with queries*, in preparation, 2005.

[39] L. Plaskota, G. W. Wasilkowski, and H. Woźniakowski, *A new algorithm and worst case complexity for Feynman-Kac path integration*, J. Comp. Phys. **164** (2000), 335–353.

[40] D. Revuz and M. Yor, *Continuous Martingales and Brownian Motion*, Springer-Verlag, 1991.

[41] L. Schmetterer, *Introduction to Mathematical Statistics*, Springer Verlag, 1974.

[42] P. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), 1484–1509.

[43] D. Simon, *On the power of quantum computation*, Proceedings of the 35th IEEE Symposium on Foundations of Computer Science, 1994, pp. 124–134.

[44] I. H. Sloan, X. Wang, and H. Woźniakowski, *Finite-order weights imply tractability of multivariate integration*, J. Complexity **20** (2004), 46–74.

[45] V. N. Temlyakov, *On approximate recovery of functions with bounded mixed derivative*, J. Complexity **9** (1993), 41–59.

[46] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-Based Complexity*, Academic Press, 1988.

[47] J. F. Traub and A. G. Werschulz, *Complexity and Information*, Cambridge University Press, 1998.

[48] J. F. Traub and H. Woźniakowski, *Path integration on a quantum computer*, Quantum Information Processing **1** (2002), 365–388.

[49] G. W. Wasilkowski and H. Woźniakowski, *Polynomial-time algorithms for multivariate linear problems with finite-order weights*, submitted.

[50] ______, *Finite-order weights imply tractability of linear multivariate problems*, J. Approximation Theory **130** (2004), 57–77.

[51] C. Wiegand, *Quantum complexity of parametric integration*, J. Complexity **20** (2004), 75–96.

[52] H. Woźniakowski, *Randomized quantum queries*, in preparation.